RASPBERRY PI BASED CYBER PHYSICAL SYSTEM FOR E-BIKE MONITORING OVER INTERNET

BY

ERIN TONG LIH XIAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I want to express my deepest gratitude to my supervisor, Dr. Teoh Shen Khang, for invaluable guidance, support, and mentorship throughout this project. Their extensive knowledge, insightful feedback, and unwavering commitment have been instrumental in shaping my research and helping me overcome the challenges of this undertaking. I am truly fortunate to have had the opportunity to work under his supervision and learn from their expertise.

I sincerely thank the lab assistants for providing me with the resources, facilities, and support necessary to complete this project. I am grateful for their assistance and the stimulating research environment they helped foster, which has contributed significantly to my academic growth and development.

On a personal note, I am deeply indebted to my family for their unwavering love, understanding, and support throughout my academic pursuits. I am especially grateful to my parents for always believing in me, for their constant encouragement, and for instilling in me the value of education and hard work.

To all those mentioned above, I extend my sincere gratitude. This accomplishment would not have been possible without your support, and I am truly thankful to each of you.

ABSTRACT

This project addresses the growing problem of e-bike theft by developing a Raspberry Pi-based Cyber Physical System (CPS) for real-time monitoring and security. With e-bike theft rates significantly exceeding motor vehicle theft and recovery rates below 15%, innovative anti-theft technologies combining prevention and recovery capabilities are urgently needed. The system integrates facial recognition using the InsightFace framework, motion detection via MPU6050 sensors, and GPS tracking with the NEO-6M module. The Raspberry Pi 4 Model B coordinates sensor data through an event-driven architecture, enabling real-time threat assessment and immediate response activation. A Logitech C270 webcam provides biometric authentication, distinguishing between authorised and unauthorised users with a confidence-based assessment. MQTT communication protocols via EMQX broker ensure reliable data transmission to a cross-platform Flutter mobile application. The mobile interface provides comprehensive monitoring through facial recognition results, interactive GPS tracking with route visualisation, and system status monitoring. Security events automatically trigger GPS tracking activation and mobile notifications for immediate theft response. Testing in Kampar, Malaysia, demonstrated reliable performance with successful unknown face detection, automatic security response activation, and accurate location tracking. The system achieved effective integration of biometric authentication, motion sensing, and location tracking into a unified security platform, providing automated theft detection and asset recovery capabilities essential for e-bike protection in urban environments.

Area of Study (Minimum 1 and Maximum 2): Cyber Physical System, Internet of Things

Keywords (Minimum 5 and Maximum 10): E-bike Security, Facial Recognition, Motion Detection, GPS Tracking, MQTT Communication, Mobile Application, Theft Prevention, Raspberry Pi

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Symbol	Meaning
%	Percentage
°/s	Degrees per second
cm	Centimeters
g	Gravitational acceleration
LSB/g	Least Significant Bit per gravitational acceleration
LSB/(°/s)	Least Significant Bit per degree per second
mA	Milliamperes
MHz	Megahertz
s	Seconds
V	Volts

LIST OF ABBREVIATIONS

API	Application Programming Interface
ARM	Advanced RISC Machine
CPS	Cyber Physical System
CPU	Central Processing Unit
EMQX	MQTT Broker Platform
GPIO	General Purpose Input/Output
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HOG	Histogram of Gradients
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
IoT	Internet of Things
LBPH	Local Binary Patterns Histograms
LTE	Long Term Evolution
MPU6050	Motion Processing Unit (6-axis sensor)
MQTT	Message Queuing Telemetry Transport
NEO-6M	GPS Module Model
OpenCV	Open Source Computer Vision Library
PIR	Passive Infrared
QoS	Quality of Service
SDK	Software Development Kit
SMS	Short Message Service
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
4G	Fourth Generation

Chapter 1
Introduction

This chapter presents the background and motivation of our research, our contributions to the field, and the thesis outline.

1.1 Problem Statement and Motivation

The rapid adoption of electric bicycles (e-bikes) as a sustainable urban transportation solution has significantly increased theft incidents, creating substantial economic losses and undermining efforts to promote eco-friendly mobility. This research addresses the critical problem of increasing e-bike theft rates and the inadequacy of current anti-theft solutions, which leave e-bike users vulnerable to substantial financial losses and discourage the adoption of sustainable transportation alternatives.

Recent studies have revealed the staggering magnitude of bicycle and e-bike theft in North America, with approximately 2.4 million adult bicycles stolen annually in the United States alone, representing a rate of 709.6 bicycles per 100,000 people per year [1]. This rate significantly exceeds motor vehicle theft rates, highlighting the severity of the problem, with the total annual economic impact of bicycle theft in the US estimated at $1.4 billion based on average stolen bicycle values [1]. The problem disproportionately affects vulnerable populations, as individuals with household incomes below $50,000 experience higher bicycle theft rates compared to those with higher incomes, and certain racial and ethnic minorities face elevated theft risks [1]. E-bikes are desirable targets for thieves, accounting for 12% of all stolen bicycles despite representing a much smaller portion of the bicycle market [2]. Their higher value, advanced technology, expensive batteries, and ease of resale make e-bikes especially vulnerable to theft, resulting in disproportionately higher theft rates than traditional bicycles [2], [3]. Recovery rates for stolen bicycles remain discouragingly low, with recent data showing only a 15% recovery rate, an improvement from previously reported 5%, meaning the vast majority of stolen bicycles are never recovered [2]. Universities and urban areas experience particularly high theft rates, with academic institutions conducting dedicated surveys to understand and address bicycle theft patterns affecting their communities [4].

Despite the growing threat of e-bike theft, current security measures have proven inadequate in providing comprehensive protection. Traditional mechanical locks, including U-locks, chains, and cable locks, remain vulnerable to modern theft techniques [5]. The vulnerability of e-bikes is exacerbated by their portability and the lucrative black market for stolen e-bikes, which can be quickly sold with altered serial numbers [6]. The situation is further complicated by inadequate secure parking infrastructure, as many areas still lack e-bike-specific parking solutions with appropriate security features [6]. The absence of standardised, effective anti-theft measures leaves e-bike owners vulnerable, particularly when parking in public spaces, workplaces, and residential areas where theft occurs.

The convergence of rising e-bike adoption, increasing theft rates, and inadequate security solutions creates an urgent need for innovative anti-theft technologies. This research is motivated by the potential to develop an integrated security system that combines multiple protection mechanisms, including biometric authentication, motion detection, and GPS tracking, to reduce e-bike theft and improve recovery rates significantly. By addressing both the prevention and recovery aspects of e-bike security through advanced cyber physical system integration, this project aims to implement the sustainable growth of e-bike usage as a viable urban transportation solution, protect users' investments, and promote equitable access to green mobility options.

1.2 Objectives

The objectives of the project aim to develop a comprehensive cyber physical security system for e-bike monitoring and theft prevention through three main objectives. The first objective is to design and implement a Raspberry Pi-based cyber physical system that integrates multiple sensing modalities, including motion detection, facial recognition, and GPS tracking capabilities, for comprehensive e-bike security monitoring. The second objective is to create an intelligent multi-modal security system that combines face recognition for user authentication, accelerometer-based movement detection for theft prevention, and automatic GPS tracking activation during security events to provide real-time location monitoring and recovery assistance. The third objective is to develop a cross-platform mobile application interface using the Flutter framework that enables users to remotely monitor system status, control GPS tracking functions, view real-time security alerts, and visualise location data through an intuitive and user-friendly interface.

1.3 Project Scope and Direction

This project includes developing hardware and software components for a comprehensive e-bike security monitoring system. The hardware scope includes integrating a single-board computer as the central processing unit, inertial measurement unit sensors for motion detection, a camera module for facial recognition capabilities, a GPS module with an external antenna for location tracking, and a power management system using a rechargeable battery configuration. The software scope covers the development of Python-based embedded system software implementing multi-threaded processing for concurrent sensor data handling, face recognition framework integration for real-time user authentication, movement detection algorithms with calibration capabilities, a GPS tracking system with automatic activation triggers, MQTT communication protocol implementation for reliable data transmission, and a cross-platform mobile application providing comprehensive system control and monitoring interfaces. The system integration scope involves implementing an event-driven architecture supporting real-time threat detection and response, establishing multi-broker communication connectivity for network reliability, and creating comprehensive user interfaces providing face detection monitoring, GPS tracking visualisation, and system status information. However, the system is designed for demonstration and prototype purposes, with testing conducted in controlled environments, and certain limitations exist, such as GPS functionality being dependent on satellite reception quality and face recognition accuracy being affected by environmental lighting conditions.

1.4 Contributions

The project has several significant contributions to e-bike security and cyber physical systems. The technical contributions include the development of an integrated multi-modal security system that combines biometric authentication, motion detection, and GPS tracking in a single cyber physical system, addressing the limitations of existing isolated security measures, and the implementation of an efficient event-driven architecture that enables real-time threat detection and response while optimising resource utilisation for mobile deployment scenarios. The practical contributions contain the creation of a comprehensive mobile monitoring platform that provides intuitive interfaces for real-time system monitoring and control, the implementation of a reliable multi-broker MQTT communication strategy that ensures robust connectivity under challenging network conditions, and the demonstration of how readily available hardware components can be integrated to create a cost-effective e-bike security solution. The research contributions include the documentation of design principles and implementation approaches for developing integrated IoT security systems, providing a framework for similar applications, the establishment of testing methodologies and performance metrics for evaluating multi-modal security systems, and the identification of best practices for integrating multiple sensor modalities, communication protocols, and user interfaces in resource-constrained embedded systems.

1.5 Report Organization

This report has seven chapters, each addressing specific aspects of the e-bike cyber physical system development and evaluation. Chapter 1 provides the foundation of the project by presenting the problem statement and motivation behind developing an integrated e-bike security system, outlining the challenges posed by increasing e-bike theft rates and inadequacy of current anti-theft solutions, establishing clear project objectives, defining the scope of work, and highlighting the technical and practical contributions of this research. Chapter 2 presents a comprehensive analysis of existing technologies and systems relevant to e-bike monitoring and cyber physical security systems, reviewing IoT architecture frameworks, existing monitoring solutions, sensor integration approaches, authentication systems, GPS tracking technologies, Internet connectivity solutions, and user interface development methodologies while identifying research gaps and establishing theoretical foundations. Chapter 3 describes the systematic approach adopted for developing the cyber physical system, including the iterative agile development methodology and event-driven architecture design, presenting technology selection rationale for hardware and software components, and providing detailed system architecture diagrams, use case descriptions, and activity diagrams. Chapter 4 provides detailed technical specifications of the system architecture, including comprehensive system block diagrams, component specifications, circuit designs, and interaction protocols, explaining the integration approach for sensors, processing units, communication modules, and power management systems. Chapter 5 documents the practical realisation of the system design, covering hardware setup procedures, software configuration, system integration processes, operational demonstrations, implementation challenges, and their solutions. Chapter 6 presents comprehensive testing methodologies, performance metrics, experimental results, and critical analysis of the system's effectiveness, evaluating the achievement of project objectives through systematic testing and providing comparative analysis with existing solutions. Chapter 7 summarises the key achievements of the project, consolidates research findings, reflects on contributions to e-bike security technology, and provides recommendations for future enhancements and research directions in cyber physical security systems.

Chapter 2
Literature Review

2.1 IoT Architecture and Cyber Physical System

Cyber Physical Systems (CPS) represent the convergence of computational elements with physical processes, creating intelligent systems capable of monitoring, controlling, and optimizing real-world operations through Internet connectivity. In the context of e-bike monitoring, CPS architecture enables seamless integration of sensors, processing units, and communication modules to create comprehensive monitoring and theft prevention systems.

Bisen et al. [7] developed a comprehensive bike monitoring system that exemplifies CPS principles through integration of physical sensors (GPS, camera) with computational processing (facial recognition algorithms) and Internet connectivity (GSM/mobile app communication). Their system architecture demonstrates how CPS enables real-time data collection from physical bike components while providing remote monitoring capabilities essential for e-bike theft prevention. The research emphasizes that CPS systems provide scalable integration of computational and physical elements, enabling continuous monitoring and automated response capabilities.

Yusro et al. [8] advanced CPS implementation in motorcycle monitoring through their Mo-SSeS system, utilizing Raspberry Pi as the central cyber component integrated with physical sensors (camera, GPS, starter switch) through Internet connectivity (Telegram). Their work demonstrates how CPS facilitates remote monitoring accessibility while maintaining real-time response to physical system changes. The study achieved significant hardware integration optimization, indicating the maturation of CPS implementation techniques in vehicle monitoring applications.

The foundational CPS architecture is further validated by Rajawat et al. [9], who implemented theft detection using Raspberry Pi with emphasis on cyber physical integration and system automation. Their research highlights how CPS systems bridge the gap between physical vehicle components and digital monitoring capabilities while reducing human intervention requirements. Similarly, Cortez et al. [12] developed the CYPHER monitoring system, demonstrating CPS capability to integrate multiple physical sensors (camera, motion detection) with cyber components (cloud storage, email alerts) into a unified monitoring architecture.

Mazlan et al. [16] contributed to CPS understanding through their surveillance system implementation using Raspberry Pi with instant notification capabilities. Their work emphasizes how CPS technology creates platforms for real-time physical-to-digital data transformation in monitoring applications, while enabling cost-effective solutions through open-source integration approaches.

However, current CPS implementations in e-bike monitoring face challenges in real-time processing requirements, network reliability, and cyber physical synchronization. The diverse integration approaches across different studies indicate a need for unified CPS frameworks specifically designed for e-bike monitoring applications with standardized cyber physical interfaces.

2.2 E-bike Specific Monitoring and Security Systems

E-bike monitoring systems require specialized approaches that address unique characteristics of electric bicycles, including battery management, motor control integration, theft vulnerability, and user authentication challenges. Unlike traditional bicycles, e-bikes represent valuable assets requiring sophisticated monitoring due to their electronic components and higher theft potential.

Bisen et al. [7] developed a monitoring system specifically addressing bike security challenges through multi-modal sensing and authentication. Their approach incorporates e-bike specific features including speed monitoring integration with bike electronics, helmet detection for safety compliance, and GPS tracking optimized for bicycle mobility patterns. The system's three-tier authentication (facial recognition, passcode, SMS) addresses e-bike specific security challenges where traditional key-based systems prove inadequate due to the electronic nature of e-bike components.

While Yusro et al. [8] focused on motorcycles, their Mo-SSeS system provides relevant insights for e-bike monitoring through integration with vehicle electrical systems. Their approach of monitoring Capacitor Discharge Ignition (CDI) system status and implementing dual power management strategies offers applicable concepts for e-bike battery and motor monitoring. The integration with existing vehicle electronics demonstrates approaches relevant for e-bike motor controller and battery management system integration.

The broader context of vehicle anti-theft systems [11] provides foundational understanding of general security approaches that can be adapted for e-bike applications. Traditional vehicle security systems demonstrate the evolution from mechanical locks to electronic authentication systems, highlighting the progression toward integrated monitoring approaches. These general vehicle security principles establish the framework for specialized e-bike implementations, though requiring significant adaptation to address the unique characteristics of electric bicycles including their dual nature as both mechanical and electronic systems.

Cabalquinto et al. [17] addressed bicycle-specific monitoring challenges through integrated lock mechanisms with advanced motion detection, though focused on traditional bicycles. Their system's sophisticated movement classification achieving 93% accuracy in distinguishing authorized from unauthorized activities provides foundational approaches applicable to e-bike monitoring. The integration of gyroscope and accelerometer sensors specifically targets bicycle theft scenarios that equally apply to e-bikes, including lifting, tampering, and unauthorized movement detection.

The comparison between general vehicle security systems [11], traditional bicycle implementations, and e-bike specific requirements reveals critical differences. General vehicle systems focus primarily on ignition and access control, while e-bikes benefit from existing electrical system integration opportunities and require battery status monitoring, motor performance tracking, and electronic component security. However, e-bikes also face increased theft risk due to higher value and require more sophisticated monitoring to protect both mechanical and electronic components compared to traditional vehicles.

2.3 Sensor Integration and Abnormal Reading Detection

Sensor integration forms the physical sensing layer of cyber physical systems for e-bike monitoring, with advanced sensor technologies enabling detection of abnormal conditions, theft attempts, and unauthorized usage patterns. The evolution from basic sensors to intelligent sensor fusion systems enables sophisticated condition monitoring and threat assessment capabilities.

Cabalquinto et al. [17] achieved breakthrough results in abnormal condition detection through implementation of inertial measurement unit (IMU) sensors combining accelerometer and gyroscope data. Their system successfully detects and classifies abnormal movements including lifting, tampering, and unauthorized handling with 93% accuracy. The research demonstrates sophisticated sensor calibration techniques essential for reliable abnormal reading detection, including offset correction achieving sensitivity scale factors of 16,384 LSB/g for accelerometers and 131 LSB/(°/s) for gyroscopes.

The abnormal condition classification algorithm implemented by Cabalquinto et al. [17] utilizes threshold-based detection with comprehensive validation through confusion matrix analysis. Their testing methodology involved 150 trials across five condition categories, achieving F1 scores ranging from 0.888 to 1.0, demonstrating reliable discrimination between normal and abnormal e-bike conditions. This approach provides foundational methodologies for e-bike monitoring systems requiring detection of theft attempts and unauthorized usage.

Traditional sensor approaches, as demonstrated by Rajawat et al. [9], rely primarily on PIR (Passive Infrared) sensors for basic motion detection. While cost-effective, these systems lack the precision required for sophisticated abnormal condition assessment in e-bike monitoring applications. The comparison between PIR-based and IMU-based systems reveals significant advantages in false alarm reduction and condition assessment accuracy when using advanced sensor technologies.

Mazlan et al. [16] implemented sensor integration through wireless sensor networks, demonstrating advantages of distributed sensing approaches for comprehensive monitoring coverage. Their work highlights how networked sensor approaches can provide distributed abnormal condition detection while maintaining cost-effectiveness through strategic sensor placement throughout the monitored system.

Bisen et al. [7] integrated multiple sensor types including GPS for location abnormalities, camera for visual verification, and speed sensors for usage pattern monitoring. Their multi-sensor approach demonstrates how sensor fusion can provide comprehensive abnormal condition detection covering location, movement, and usage pattern anomalies essential for e-bike monitoring applications.

However, advanced sensor integration systems face challenges in environmental adaptation, power consumption management, and real-time processing requirements. The need for continuous sensor monitoring while maintaining extended battery operation requires careful optimization of sensor selection and data processing algorithms for e-bike applications.

2.4 Camera-based Authentication and Visual Monitoring

Camera-based authentication in e-bike monitoring systems provides visual verification capabilities that enhance security through user identification while enabling documentation of theft attempts and unauthorized access events. Visual monitoring capabilities are essential for e-bike systems due to the high value and theft vulnerability of electronic components.

Bisen et al. [7] implemented camera-based facial recognition as the primary authentication method for e-bike access control, utilizing webcam-based face capture and comparison against stored user databases. Their system provides automated e-bike activation for recognized users while capturing images of unauthorized access attempts for owner notification. The integration of facial recognition with GPS location and photo transmission demonstrates comprehensive visual monitoring capabilities essential for e-bike theft prevention and recovery.

Zuma et al. [13] advanced camera-based authentication through integration with IoT systems using Raspberry Pi and OpenCV framework. Their research utilized Haar Cascade Classifier and Histogram of Gradients (HOG) algorithms for face detection, achieving 84% overall detection accuracy. The study demonstrates practical implementation of visual authentication in monitoring systems with real-time processing capabilities suitable for e-bike applications requiring immediate user verification.

Chandrakasan et al. [15] contributed sophisticated visual recognition algorithms through implementation of Local Binary Patterns Histograms (LBPH) for face recognition on Raspberry Pi with OpenCV. Their research addresses computational constraints inherent in embedded systems while maintaining real-time processing capabilities. The comprehensive methodology covering face detection, data gathering, training, and recognition phases provides practical guidance for implementing camera-based authentication in e-bike monitoring systems.

Yusro et al. [8] implemented camera systems for theft documentation, capturing images of unauthorized access attempts and transmitting them through Internet connectivity to vehicle owners. Their Pi-camera integration demonstrates how visual monitoring can provide evidence collection capabilities essential for e-bike theft investigation and recovery processes.

The visual monitoring capabilities extend beyond authentication to include abnormal activity documentation, theft attempt recording, and usage pattern analysis. Camera integration enables e-bike monitoring systems to provide visual evidence of security events while supporting user authentication and access control functions.

However, camera-based systems face challenges, including lighting sensitivity, weather protection requirements, privacy concerns, and processing power limitations in embedded e-bike monitoring applications. The balance between visual monitoring capabilities and user privacy presents ongoing challenges requiring careful system design and implementation.

2.5 GPS Integration and Location-based Monitoring

GPS integration provides essential location tracking capabilities for e-bike monitoring systems, enabling theft recovery, route monitoring, geofencing, and location-based security features. The accuracy and reliability of GPS systems directly impact the effectiveness of location-based monitoring and theft recovery capabilities.

Bisen et al. [7] implemented GPS technology for precise e-bike location determination, integrating GPS data with mobile communication for real-time location transmission to e-bike owners. Their system demonstrates GPS integration for theft recovery and location monitoring, providing coordinates through mobile application interfaces and SMS notifications. The research emphasizes how GPS technology enables “find my e-bike” functionality essential for theft recovery in urban environments where e-bikes are commonly stolen.

Yusro et al. [8] achieved superior GPS implementation through NEO-M8N GPS module integration, demonstrating location accuracy within 1.124 to 7.038 meters when validated against Google Maps references. Their comprehensive testing methodology with statistical analysis of coordinate differences provides benchmarks applicable to e-bike monitoring systems requiring reliable location tracking. The achieved accuracy levels are sufficient for e-bike theft recovery and route monitoring applications.

Advanced GPS tracking applications incorporating Kalman filter methodologies [10] provide enhanced accuracy through predictive filtering and noise reduction techniques. Kalman filter implementation addresses GPS signal noise and trajectory prediction challenges commonly encountered in urban environments where e-bikes operate. The filtering algorithms enable improved location accuracy by combining GPS measurements with motion prediction models, particularly valuable for e-bike monitoring systems requiring precise location tracking despite signal interference from urban infrastructure.

The infrastructure study [14] provides critical insights into environmental factors affecting GPS accuracy in cycling environments. The research examines how cycle path characteristics, including surface types (stone pavement, asphalt concrete pavement), tree coverage, and building density impact GPS signal reception and coordinate accuracy. The study demonstrates that GPS trajectory discrepancies occur in areas with significant tree or building coverage due to Signal Multipath phenomenon, where satellite signals are reflected off obstacles, thereby altering GPS coordinates. These findings are particularly relevant for e-bike monitoring systems operating in diverse urban environments with varying infrastructure characteristics.

The integration of Kalman filter techniques [10] with infrastructure-aware GPS processing [14] offers potential solutions for maintaining location accuracy across diverse e-bike operating environments. The predictive filtering capabilities can compensate for GPS signal disruptions in challenging infrastructure conditions, while the infrastructure analysis provides understanding of environmental factors affecting GPS performance.

The GPS integration enables advanced monitoring features including route tracking for usage analysis, geofencing for theft detection when e-bikes are moved outside authorized areas, and location-based alerts for owners. These capabilities are particularly valuable for e-bike monitoring due to the mobility patterns and theft vulnerability of electric bicycles in urban environments.

2.6 Internet Connectivity and Remote Monitoring Systems

Internet connectivity enables remote monitoring capabilities essential for e-bike monitoring systems, providing real-time data access, remote control functions, and instant notification delivery to e-bike owners regardless of physical location. The choice of Internet connectivity solutions significantly impacts system reliability and monitoring effectiveness.

Bisen et al. [7] implemented GSM-based Internet connectivity for mobile network communication, enabling SMS-based notifications and mobile application-based remote monitoring. Their system utilizes cellular data connectivity for real-time information transmission including location data, security alerts, and system status updates. The GSM integration demonstrates reliable Internet connectivity for e-bike monitoring applications in areas with cellular network coverage.

Yusro et al. [8] advanced Internet connectivity through Telegram bot integration, providing sophisticated command-based remote monitoring and control capabilities. Their system demonstrates how Internet-based messaging platforms can serve as effective interfaces for e-bike monitoring system control, offering cross-platform compatibility and user-friendly interfaces. The Telegram integration enables image transmission, location sharing, and remote system monitoring through Internet connectivity.

Zuma et al. [13] validated Internet-based monitoring effectiveness through Telegram integration for real-time security alerts and image transmission. Their research demonstrates how Internet connectivity enables rich media communication essential for comprehensive e-bike monitoring, including photo transmission of theft attempts and real-time location data sharing.

Mazlan et al. [16] implemented dual Internet connectivity approaches through both email and instant messaging, providing redundancy in communication channels essential for reliable e-bike monitoring. Their system architecture demonstrates the importance of multiple Internet communication pathways to ensure reliable monitoring data delivery under various network conditions.

Cortez et al. [12] developed cloud-based Internet connectivity for e-bike monitoring through Google Drive integration, demonstrating how cloud services can provide reliable data storage and access for monitoring systems. Their approach enables remote data access and system monitoring through standard Internet protocols.

However, Internet connectivity implementation faces challenges in network reliability, data costs, and connectivity coverage, particularly relevant for e-bike applications that may operate in areas with limited network coverage. The dependence on Internet connectivity requires careful consideration of offline monitoring capabilities and data buffering strategies.

2.7 GUI Development and Information Visualization

Graphical User Interface (GUI) development and information visualization form the human-machine interface layer of e-bike monitoring systems, enabling users to access monitoring data, control system functions, and visualize e-bike status information through intuitive interfaces. Effective GUI design is essential for user adoption and system effectiveness.

Bisen et al. [7] developed mobile application interfaces for e-bike monitoring system control, providing GUI elements for location tracking, system status monitoring, and security feature control. Their interface design demonstrates how mobile GUIs can provide comprehensive monitoring capabilities including real-time location display, security system activation, and alert management. The integration of GPS mapping with security controls shows effective information visualization for e-bike monitoring applications.

Yusro et al. [8] implemented Telegram-based GUI interfaces through bot commands, providing text-based control interfaces for e-bike monitoring functions. While primarily text-based, their approach demonstrates how messaging platform interfaces can provide effective user interaction for monitoring system control, though limited in graphical data visualization capabilities.

Cortez et al. [12] developed web-based GUI interfaces for security system monitoring, providing browser-based access to monitoring data and system controls. Their approach demonstrates how web-based interfaces can provide cross-platform monitoring access while supporting rich data visualization including live video feeds and system status displays.

The GUI development for e-bike monitoring systems must address unique requirements including mobile device compatibility for on-the-go monitoring, real-time data visualization for immediate status assessment, and intuitive control interfaces for emergency situations. The visualization of sensor data, location information, and security status requires careful interface design to provide clear, actionable information to users.

However, GUI development faces challenges in cross-platform compatibility, real-time data display performance, and user interface complexity management. The balance between comprehensive functionality and interface simplicity presents ongoing challenges in e-bike monitoring system GUI development, particularly for mobile device interfaces with limited screen space.

2.8 Lessons Learned and Design Principles

The comprehensive analysis of existing research reveals valuable lessons and design principles that inform effective cyber physical system development for e-bike monitoring applications. These insights provide constructive guidance for implementing reliable and user-friendly monitoring systems.

Multi-Modal Authentication Effectiveness

The successful implementation by Bisen et al. [7] demonstrates that multi-modal authentication significantly enhances system reliability through redundancy. Their three-tier approach (facial recognition, passcode, SMS) achieved robust user authentication by providing fallback options when individual methods fail due to environmental conditions or technical issues. This principle indicates that e-bike monitoring systems should incorporate multiple authentication mechanisms to ensure consistent user access across diverse operational scenarios.

Hardware Integration Optimization

Yusro et al. [8] successfully demonstrated hardware minimization principles by achieving superior performance with reduced physical complexity compared to previous implementations. Their approach of integrating monitoring components directly into existing vehicle systems (speedometer integration) provides the design principle that effective e-bike monitoring requires seamless integration with existing e-bike electronics rather than standalone system approaches.

Sensor Fusion for Accuracy Enhancement

The outstanding 93% accuracy achieved by Cabalquinto et al. [17] through IMU sensor fusion establishes the principle that combining complementary sensor technologies yields superior performance compared to single-sensor approaches. Their systematic calibration methodology and threshold-based classification demonstrate that rigorous sensor calibration and intelligent data processing are essential for reliable abnormal condition detection in e-bike monitoring applications.

Real-Time Processing Optimization

Chandrakasan et al. [15] and Zuma et al. [13] both successfully implemented real-time facial recognition on resource-constrained embedded systems, establishing the principle that careful algorithm selection and optimization enable sophisticated processing capabilities on embedded platforms. Their work demonstrates that OpenCV framework with optimized algorithms can provide real-time biometric authentication suitable for e-bike monitoring applications.

Communication System Reliability

The successful Telegram integration by both Yusro et al. [8] and Zuma et al. [13] establishes the design principle that modern messaging platforms provide reliable, user-friendly communication channels for monitoring system alerts and control. The cross-platform compatibility and rich media support of messaging platforms offer superior user experience compared to traditional SMS-based communication systems.

User Interface Design Principles

The successful mobile application implementations across multiple studies [7],[12],[16] demonstrate that effective user interfaces must prioritize real-time data visualization, intuitive control mechanisms, and cross-platform compatibility. The integration of mapping interfaces with monitoring controls shows that effective e-bike monitoring systems require comprehensive GUI design that presents complex information in accessible formats.

2.9 Literature Map

Table 2.1 Literature Map

Ref	**Hardware**	**Software**	**Limitations**
[7]	• Raspberry Pi • Arduino • GPS module • GSM module • USB webcam	• Face recognition (AI algorithm) • Local Binary Pattern (LBP) • Android mobile app • SMS system	• Complex hardware setup • Requires GPS antenna for accuracy • Memory management issues

	• Relay circuits • Keypad		
[8]	• Raspberry Pi 3 Model B+ • NEO-M8N GPS • Pi-Camera • 18650 lithium batteries • LM338 regulator • Relay modules	• Telegram bot app • Python programming • IoT integration • Real-time monitoring	• GPS accuracy (1-6 meter variance) • Physical size constraints • Voltage regulation issues
[9]	• Raspberry Pi 3 • PIR sensor • Pi Camera • Wi-Fi adapter • Power supply (5.1V)	• Python programming • OpenCV library • Local Binary Pattern (LBP) • Motion detection algorithm	• Motion detection only • No identity verification • Lighting dependent • No theft prevention capability
[10]	• Raspberry Pi 3 • Low-cost GPS receiver • Wi-Fi/hotspot device • USB components	• Kalman Filter algorithm • Python programming • Google Maps integration • MQTT protocol	• Weather affects accuracy • Speed impacts precision • Parameter optimization needed
[11]	• Raspberry Pi 4 Model B • GPS, GSM modules • Fingerprint sensor • Pi Camera, keypad, tilt sensor	• Face recognition • Fingerprint verification • Password system • SMS communication	• GSM-only (no internet) • Complex multi-input system • Crankshaft sensor dependency
[12]	• Raspberry Pi 3B • Pi Camera • IR LED • CoHeatsint & fan	• OpenCV & Python Flask • HAAR cascade classifier • Google Drive storage • Email notifications	• Lighting affects detection • Overheating concerns • Internet dependency
[13]	• Raspberry Pi 4 Model B	• Python & OpenCV	• 84% overall accuracy

	• PIR sensor, webcam • LED indicators • 5V power supply	• Telegram bot API • HOG facial recognition	• False identification issues
[14]	• Raspberry Pi 4 Model B • IMU (GY-521/MPU6050) • GPS (NEO-6M) • Camera V3 Standard • Hall Effect sensor (KY-003) • Arduino Uno	• Raspbian OS • Python • OpenCV • QGIS	• Single e-bike configuration • limited to cobblestone/asphalt surfaces
[15]	• Raspberry Pi • PiCam camera module • basic hardware setup	• OpenCV, Python • Haar Cascade classifier • LBPH face recognizer • Raspbian OS	• Resource constraints of Pi • lighting conditions affect accuracy • single bicycle configuration
[16]	• Raspberry Pi 4 Model B • PIR motion sensor • Camera Module V2	• Raspbian OS • Python • Telegram API • Email API • MQTT protocol • Thonny IDE	• Requires internet connectivity • complex setup for non-technical users • rural area limitations
[17]	• MPU-6050 IMU • voltage generator • RFID module • cable lock detector, audible alarm • GPS tracker	• Arduino IDE • movement classification algorithms • SMS notification system	• Cable lock sold separately • locks can be cut with tools

Chapter 3
System Methodology/Approach

3.1 Development Methodology

The project adopted an iterative agile development methodology designed to accommodate the complex nature of cyber physical systems integration, where hardware and software components must seamlessly interact to provide reliable security monitoring capabilities. This development approach was selected due to several key advantages that align with the project's requirements for rapid prototyping, continuous validation, and flexible adaptation to emerging technical challenges. The iterative development process was strategically divided into multiple cycles, with each iteration focusing on specific system components to ensure thorough testing and validation before integration into the complete system architecture. The first iteration concentrated on basic sensor integration and data collection mechanisms, establishing foundational hardware interfaces and sensor calibration procedures. Subsequent iterations addressed communication system implementation, mobile application development, and comprehensive system integration with performance optimisation.

The agile methodology facilitated rapid prototyping capabilities that proved essential for validating design concepts in real-world conditions, allowing each sensor module and software component to be individually tested and validated before integration into the complete system architecture. This approach significantly reduced development risks and enabled early identification of potential integration issues that could impact system performance. Continuous testing and validation throughout all development phases ensured that hardware malfunctions, software bugs, and integration conflicts were identified and resolved early in the development cycle, preventing costly redesign efforts and maintaining project timeline adherence. The development process followed four distinct phases: requirements analysis and technology assessment involving a comprehensive analysis of e-bike security requirements and evaluation of available sensor technologies, system architecture design focusing on hardware component selection and software architecture planning, implementation and integration encompassing individual component development and comprehensive system testing, and evaluation and optimisation involving system performance assessment and resource utilisation optimisation.

3.2 Cyber Physical System Architecture Approach

3.2.1 Event-Driven Architecture

The system adopts an event-driven architecture to ensure real-time responsiveness to security threats while maintaining optimal resource utilisation throughout extended monitoring periods. This architectural approach proves remarkably effective for e-bike monitoring applications where immediate reaction to specific events is critical for successful theft prevention and detection capabilities. The architecture identifies four primary event sources that drive comprehensive system behaviour: movement detection events originating from the MPU6050 accelerometer and gyroscope sensor when unauthorised physical manipulation exceeds calibrated thresholds, face recognition events triggered by the camera system upon detecting unknown individuals within the monitoring frame, timer-based events occurring at regular intervals for routine data transmission and system health monitoring, and command events initiated through the mobile application interface to enable remote system control and configuration management.

The central event processing mechanism operates on the Raspberry Pi platform by implementing a multi-threaded event handler that processes concurrent events without blocking critical system operations. Each event type undergoes specific validation and filtering procedures to prevent false triggers while optimising system response accuracy and reliability. The processing layer maintains sophisticated event queues to handle simultaneous event occurrences and implements priority-based scheduling algorithms where security-related events receive higher precedence than routine monitoring tasks. This ensures critical security threats are addressed immediately while maintaining overall system stability and performance. The architecture implements structured response protocols that are automatically triggered by processed events, where movement detection events initiate GPS tracking sequences and sensor recalibration procedures, unknown face detection events trigger countdown mechanisms with emergency response preparation, timer events facilitate regular MQTT data transmission and system status updates, and command events execute immediate system state changes and configuration modifications as requested through the mobile interface.

The system utilises the MQTT protocol for efficient event propagation between distributed system components, structuring event data into topic-specific messages that enable selective subscription patterns and reduce network overhead. The mobile application subscribes to relevant event topics, receiving real-time notifications for security incidents, system status changes, and GPS tracking updates through this publish-subscribe model that ensures loose coupling between system components while maintaining reliable event delivery through Quality of Service (QoS) guarantees. This event-driven approach provides several critical advantages for e-bike security monitoring, including real-time responsiveness that ensures immediate threat detection and response, resource efficiency achieved through reactive processing where system resources are allocated only when events require attention, and natural scalability that allows additional sensor types or response mechanisms to be integrated through new event handlers without modifying existing system logic.

3.2.2 Real-Time Processing Strategy

The multi-threaded processing strategy ensures concurrent handling of sensor data acquisition, event processing, and communication management without compromising system performance or security monitoring effectiveness. The central monitoring thread operates continuously to process movement detection through accelerometer data analysis and face recognition through camera input processing, while simultaneously managing event correlation and system state coordination. This primary thread implements sophisticated filtering algorithms and threshold comparisons to distinguish genuine security threats from environmental noise, ensuring accurate threat assessment while minimising false alarm occurrences. The movement detection processing applies calibrated baseline adjustments and consecutive detection requirements to verify authentic security events before triggering response protocols.

A dedicated GPS tracking thread activates automatically when security events are detected, operating independently to collect location data, generate route maps, and transmit tracking information without interrupting ongoing security monitoring functions. This thread manages GPSD communication, coordinate validation, and map generation processes while maintaining precise location accuracy through systematic error correction and trajectory analysis. The GPS processing thread operates with configurable duration limits and automatic termination conditions to optimise power consumption while ensuring comprehensive location tracking during security incidents. The MQTT communication thread manages all network connectivity and message processing independently, preventing latency or temporary connectivity issues from affecting critical security event detection capabilities. This dedicated communication thread handles broker connection management, message queuing, automatic reconnection procedures, and QoS implementation to ensure reliable data transmission even under challenging network conditions.

Thread synchronisation mechanisms utilise proper locking protocols to ensure thread-safe access to shared system variables, preventing race conditions and maintaining data consistency across all system operations. The implementation employs threading locks to coordinate access to critical variables such as GPS tracking status, system operational modes, and sensor calibration parameters. System coordination threads manage event response sequencing and maintain coherent behaviour when multiple events coincide, implementing priority-based processing that ensures security events receive immediate attention. At the same time, routine maintenance tasks are deferred appropriately. This comprehensive multi-threaded architecture enables the e-bike CPS tracker to maintain constant security vigilance while efficiently managing computational resources and delivering immediate response to security incidents. This ensures reliable protection for e-bike assets across various operational environments and network conditions.

3.3 System Design Diagram

3.3.1 System Architecture Design

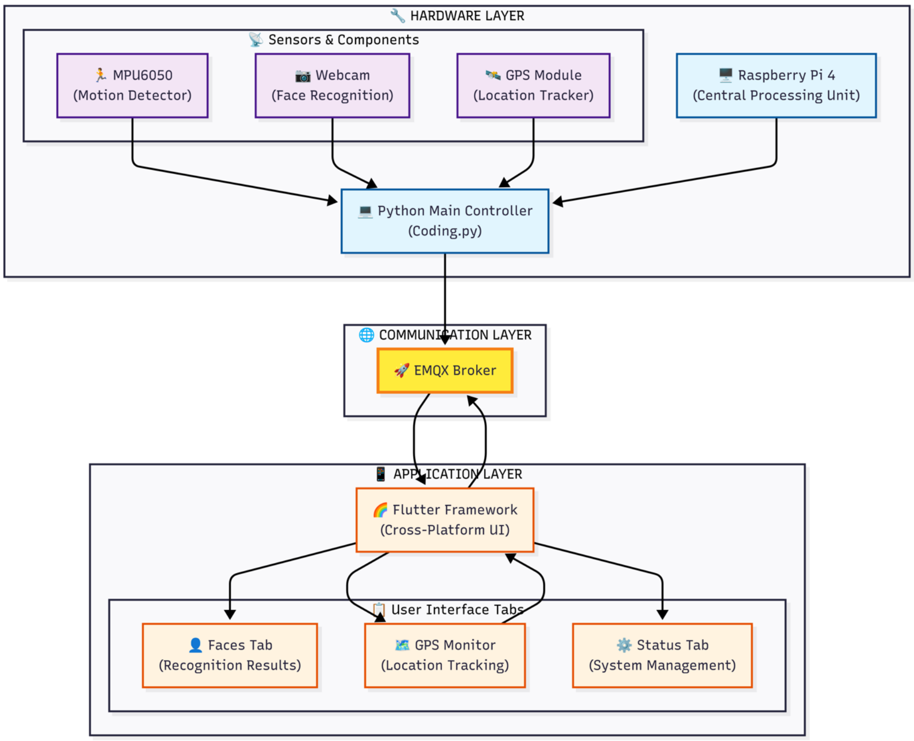


Figure 3.1 System Architecture Diagram

The E-bike CPS Tracker system employs a three-layer architecture that seamlessly integrates hardware sensors, communication protocols, and mobile application interfaces. The **Hardware Layer** forms the foundation with a Raspberry Pi 4 serving as the central processing unit, coordinating data from three critical sensors: the MPU6050 motion detector for movement analysis, Pi Camera for facial recognition capabilities, and GPS module for location tracking. All sensor data flows into the Python Main Controller (Coding.py), which acts as the system's intelligence hub, processing sensor inputs and implementing the core monitoring algorithms. The **Communication Layer** provides robust connectivity through MQTT broker redundancy, with EMQX serving as the primary broker for real-time data transmission. The **Application Layer** utilizes the Flutter framework to deliver a cross-platform mobile interface, distributing real-time system data across three specialized tabs: the Faces Tab displays facial recognition results, the GPS Monitor provides location tracking with bidirectional control capabilities for starting and stopping GPS tracking, and the Status Control tab presents comprehensive system management information. This architecture enables real-time monitoring, theft detection through unknown face recognition, movement-based alerts, and remote GPS tracking, creating a comprehensive cyber physical security system for e-bike protection.

3.3.2 Use Case and Description

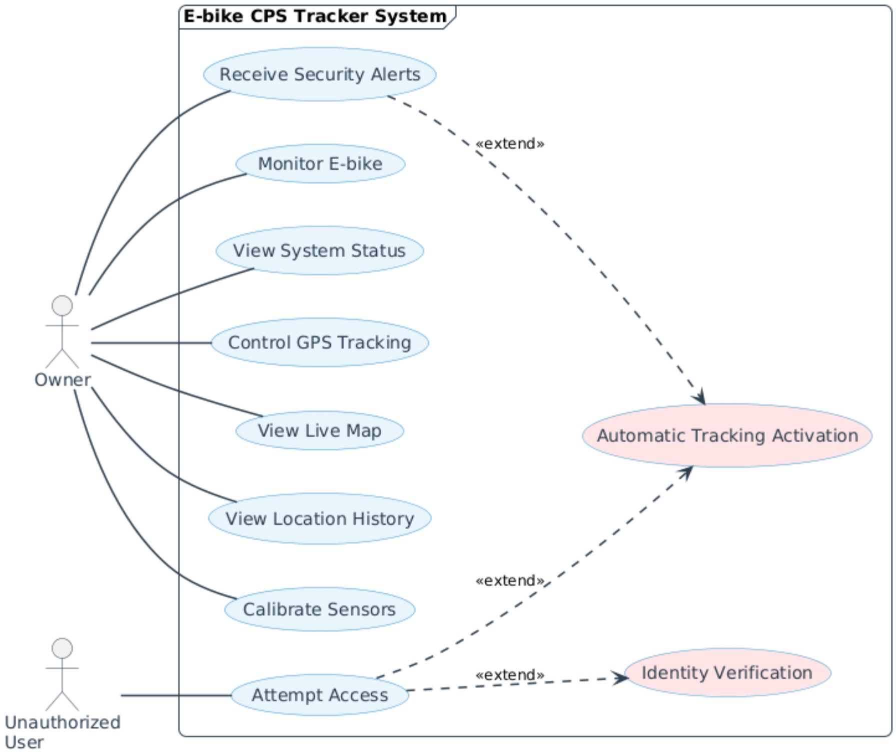


Figure 3.2 Use Case Diagram

The use case diagram illustrated in Figure 3.2 provides a comprehensive overview of the E-bike CPS Tracker system's functional requirements and user interactions. The diagram identifies two primary actors and their use cases, establishing the foundation for understanding system behaviour and user goal fulfilment throughout various operational scenarios. The system identifies two primary actors. The Owner is the legitimate e-bike user who actively manages and monitors the e-bike security system through the mobile application interface, including real-time monitoring, location tracking, system configuration, and security oversight. The Unauthorised User represents any individual who attempts unauthorised access to the e-bike, serving as an external trigger that activates the system's security detection and response capabilities.

The Owner interacts with seven primary use cases that provide comprehensive e-bike management capabilities. Monitor E-bike enables continuous oversight of the e-bike's security status and operational conditions. View System Status provides real-time information about sensor functionality, connectivity status, and overall system performance through MQTT communication. Control GPS Tracking allows manual activation and deactivation of location monitoring based on user preferences. View Live Map provides real-time geographical visualisation through GPS integration and interactive mapping. View Location History enables access to historical tracking data for pattern analysis and theft investigation. Calibrate Sensors maintains optimal sensor performance through motion detection calibration procedures.

Receive Security Alerts is a distinct use case that provides real-time communication of security events and system status updates by implementing the MQTT protocol. This contains security warnings, movement detection events, unknown face alerts, and critical system notifications, delivering automated alerts regardless of whether the Owner is actively monitoring the system. The Unauthorised User interacts with the Attempt Access use case, which triggers the system's security detection mechanisms and initiates protective protocols through automated responses.

The diagram demonstrates critical extent relationships that showcase automated security capabilities. Automatic Tracking Activation extends both Receive Security Alerts and Attempt Access, representing the system's capability to automatically initiate GPS tracking when security events are detected, including movement above calibrated thresholds or unauthorised access attempts. Identity Verification extends Attempt Access by providing automated facial recognition that distinguishes between authorised and unauthorised users, implementing a fifteen-second countdown mechanism for unknown faces before triggering security protocols.

These relationships ensure the system provides automated security responses that enhance user-initiated activities. The separation of Receive Security Alerts from Monitor E-bike reflects the distinct nature of active user engagement versus passive receipt of automated notifications, creating a cohesive security ecosystem with active user control and passive protection mechanisms.

3.3.3 Activity Diagram

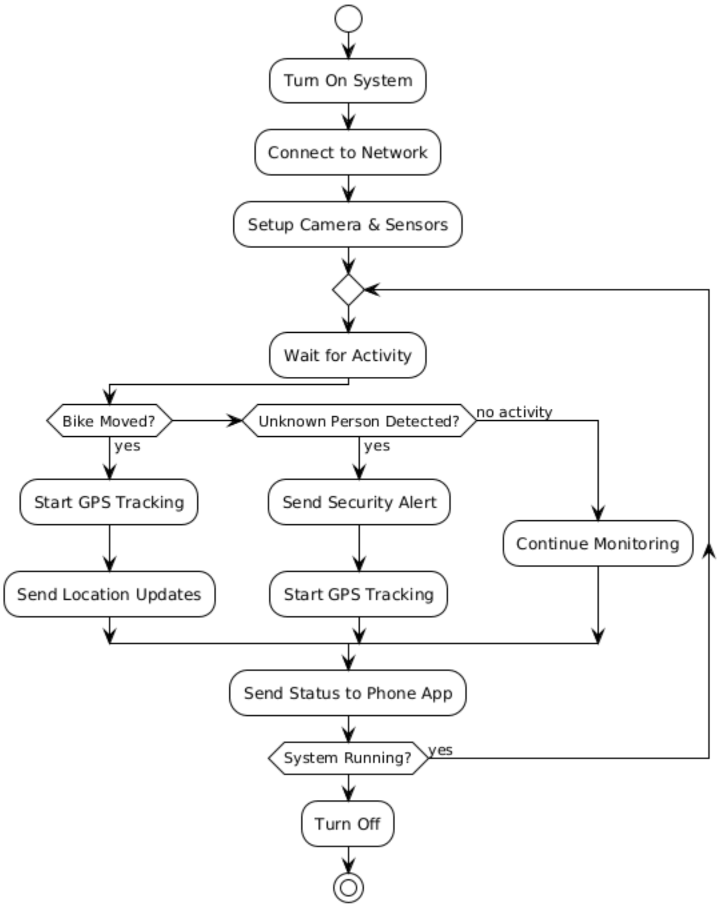


Figure 3.3 Activity Diagram

The activity diagram presented in Figure 3.3 illustrates the comprehensive operational flow of the E-bike Security System from initialization through monitoring activities to system termination. The operational flow commences with system activation through three sequential initialization activities: “Turn On System” initializes all hardware components and loads essential software modules, “Connect to Network” establishes MQTT communication channels with the mobile application and cloud services, and “Setup Camera & Sensors” calibrates the facial recognition system, motion detection sensors, and GPS module as implemented in the Python code's initialization sequence.

After initialising successfully, the system enters a continuous monitoring loop centred on parallel decision-making that evaluates two critical conditions simultaneously: bike movement detection and unknown person identification. The “Bike Moved?” decision point utilises accelerometer and gyroscope data from the MPU6050 sensor to detect unauthorised movement patterns, while the “Unknown Person Detected?” decision point processes facial recognition results from the camera system. When either condition is triggered, the system immediately initiates “Start GPS Tracking” and executes appropriate response actions “Send Location Updates” for movement detection and “Send Security Alert” for unknown face detection, implementing the automated threat response mechanisms described in the event-driven architecture.

The activity diagram demonstrates sophisticated parallel processing capabilities where the “Continue Monitoring” activity maintains constant surveillance of both movement and facial recognition systems while GPS tracking and alert transmission occur concurrently when triggered. Throughout all operational phases, the “Send Status to Phone App” activity provides continuous communication with the mobile interface through MQTT protocol, ensuring users maintain real-time awareness of system health and security events. The operational flow incorporates a “System Running?” decision point that enables graceful system shutdown through the “Turn Off” activity, allowing natural termination of the monitoring loop while ensuring all active processes complete appropriately.

Chapter 4
System Design

4.1 Technology Selection Rationale

4.1.1 Hardware Platform Selection

Raspberry Pi: The Raspberry Pi 4 model B was selected as the primary computing platform due to its balance of processing capability, GPIO availability, and power efficiency. The ARM Cortex-A72 processor provides sufficient computational power for real-time face recognition and sensor data processing while maintaining reasonable power consumption for mobile e-bike deployment.

Figure 4.1 Raspberry Pi 4 Model B

Motion Sensor: GY-521 (MPU6050 Accelerometer/Gyroscope) sensor was chosen for motion detection due to its integrated 6-axis motion sensing capability, I2C communication interface compatibility with Raspberry Pi, and proven accuracy in motion detection applications. The sensor's capability to detect linear acceleration and angular velocity enables sophisticated movement pattern analysis.

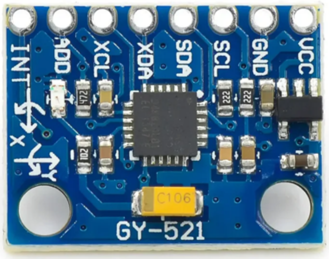


Figure 4.2 GY-521

Camera Module: The Logitech C270 webcam provides reliable USB connectivity with Raspberry Pi for face recognition applications. This webcam delivers 720p HD video with automatic light correction and plug-and-play integration. The C270's compact design and automatic focus optimise image quality across varying lighting conditions, enhancing face recognition reliability.

Figure 4.3 Logitech C270 webcam

GPS Module: The GY-NEO-6M GPS module with external active ceramic antenna provides enhanced location tracking through UART communication interface. The module utilizes the u-blox NEO-6M GPS chipset operating on standard GPS frequencies (1561 & 1575 MHz), while the external active ceramic antenna with SMA male interface significantly improves signal reception in mobile environments. This antenna configuration ensures reliable satellite acquisition even when the GPS module is enclosed within the e-bike housing, with GPSD software stack compatibility for efficient coordinate data processing.

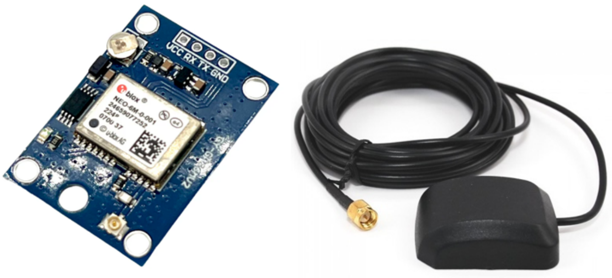


Figure 4.4 NEO-6M with antenna

Communication Module - 4G/LTE Dongle: A 4G USB dongle was selected over dedicated 4G GSM modules due to cost considerations, as professional GSM modules cost 200-500% more while providing similar connectivity capabilities. The selected dongle offers reliable cellular internet access through standard USB interface with plug-and-play Raspberry Pi compatibility, supporting both WiFi and 4G/LTE connectivity essential for MQTT communication. This cost-effective solution includes dual-band WiFi (2.4GHz/5GHz) as backup connectivity and supports multiple cellular bands for broad network compatibility while maintaining project budget feasibility.

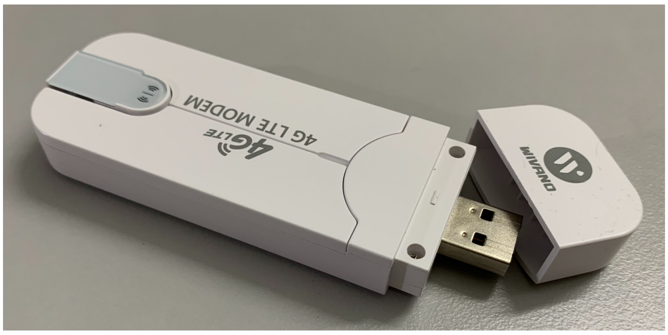


Figure 4.5 4G/LTE Dongle

Power Management System: The system uses Rakieta 18650 Li-ion batteries (12000mAh, 3.7V) with Battery Shield V8 module for mobile power. The dual-battery configuration provides 24000mAh capacity for extended operation, while the shield delivers a stable 5V output with overcharge protection and dual charging interfaces (Micro USB/Type-C).

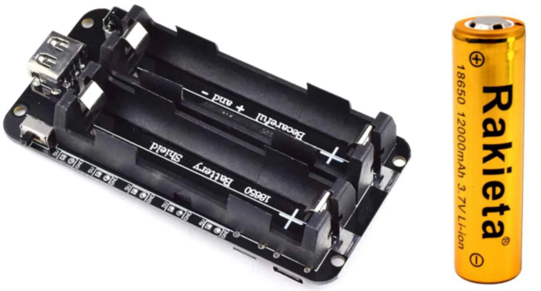


Figure 4.6 Battery Shield and Battery

4.1.2 Software Technology Selection

Python for Embedded Development: Python was selected for the embedded system development due to its extensive library support for sensor interfacing (GPIO, I2C protocols), computer vision processing (OpenCV), and machine learning applications (InsightFace). The language's rapid development capabilities align with the agile methodology requirements.

Flutter for Mobile Application: Flutter framework was chosen for cross-platform mobile application development, enabling simultaneous iOS and Android deployment from a single codebase. The framework's reactive programming model suits real-time data visualisation requirements.

MQTT Communication Protocol: MQTT was selected over HTTP-based protocols due to its lightweight nature, publish-subscribe architecture suitability for IoT applications, and Quality of Service (QoS) guarantees for critical message delivery.

4.1.3 Communication Infrastructure

EMQX Broker Strategy: The system implements EMQX public broker (broker.emqx.io) as the primary MQTT communication hub with robust retry mechanisms to ensure reliable connectivity. The implementation includes automatic broker connectivity testing and exponential backoff reconnection strategies to handle network interruptions common in mobile deployment scenarios.

Retry and Reconnection Mechanism: The system incorporates intelligent connection management, performing up to 20 reconnection attempts with exponential backoff delays ranging from 3 to 10 seconds. Each connection attempt includes network interface validation and broker reachability testing before establishing MQTT communication, ensuring optimal resource utilisation and preventing unnecessary connection attempts during network unavailability.

Quality of Service Implementation: The MQTT client supports multiple QoS levels (0, 1, 2) with automatic reconnection capabilities and connection status monitoring. The system maintains persistent connections through keep-alive mechanisms and implements graceful disconnection handling to preserve message delivery guarantees during temporary network disruptions.

Topic-Based Architecture: MQTT topic organisation separates different data types through structured naming conventions: pi_tracker/data for comprehensive sensor information, pi_tracker/status for system events and alerts, pi_tracker/gps for location coordinates, and pi_tracker/commands for bidirectional control messages. This hierarchical topic structure enables efficient message routing and selective subscription patterns while supporting scalable deployment scenarios.

4.2 System Block Diagram

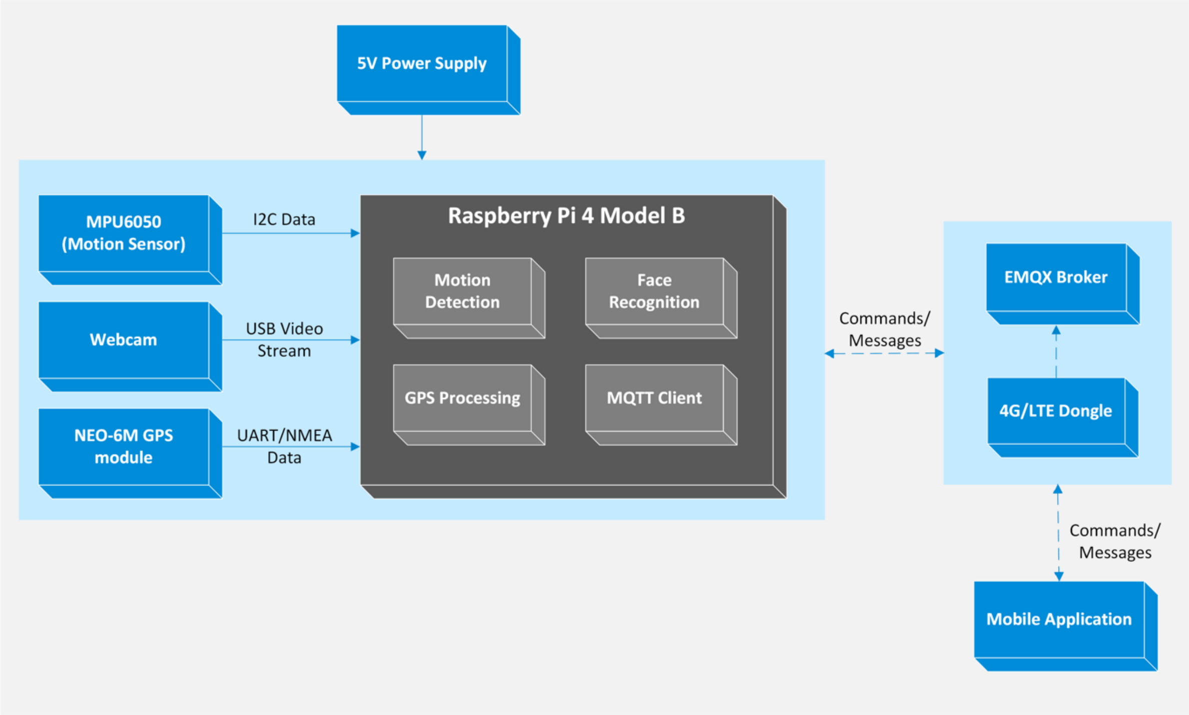


Figure 4.7 System Block Diagram

Figure 4.7 presents the system block diagram illustrating the interconnected components and data flow pathways within the e-bike CPS tracker system. The diagram shows the Raspberry Pi 4 Model B as the central processing hub, positioned at the centre with multiple input and output connections to various system components. Three primary sensors connect directly to the Raspberry Pi 4 Model B through dedicated communication interfaces. The MPU6050 motion sensor establishes a connection via I2C Data transmission, providing continuous movement and orientation information for theft detection algorithms. The Webcam connects through the USB Video Stream interface, delivering real-time video data for facial recognition processing. The NEO-6M GPS module communicates through UART/NMEA Data protocol, supplying precise location coordinates for tracking functionality.

The Raspberry Pi 4 Model B contains four internal processing modules handling incoming sensor data. The Motion Detection module processes accelerometer and gyroscope inputs from the MPU6050 sensor. The Face Recognition module analyses video frames captured by the Webcam. The GPS Processing module manages location data received from the GPS module. The MQTT Client module handles all network communications and data transmission protocols.

Power distribution flows directly from the 5V Power Supply unit to the Raspberry Pi 4 Model B, ensuring stable electrical operation for all integrated processing modules and external sensor connections. The power supply system provides the necessary voltage regulation and current capacity for continuous system operation. Network connectivity is established through two parallel pathways that enable robust communication capabilities. The MQTT Client within the Raspberry Pi connects to the EMQX Broker through bidirectional Command/Message transmission. The 4G/LTE Dongle provides internet connectivity to the EMQX Broker, ensuring reliable network access even in mobile deployment scenarios where WiFi connectivity may be unavailable.

The mobile application completes the communication chain by connecting to the EMQX Broker and exchanging commands/messages. This connection enables real-time data reception from the embedded system and allows users to send control commands back to the Raspberry Pi for system configuration and operational control. The block diagram demonstrates a centralised processing approach where all sensor data converges at the Raspberry Pi 4 Model B for analysis and decision-making. At the same time, communication flows through the EMQX Broker to enable remote monitoring and control through the mobile application interface.

4.3 System Components Specification

4.3.1 Central Processing Unit

Raspberry Pi 4 Model B serves as the primary computing platform for the entire system, providing sufficient computational power for concurrent sensor data processing, facial recognition algorithms, and communication management.

Table 4.1 Specification of Raspberry Pi

Specification	Details
Processor	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	8GB LPDDR4-3200 SDRAM
Storage	MicroSD card slot for loading operating system and data storage
Connectivity	• 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN • Bluetooth 5.0, BLE • Gigabit Ethernet • 2 × USB 3.0 ports, 2 × USB 2.0 ports
GPIO	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & Sound	• 2 × micro-HDMI ports (up to 4Kp60 supported) • 2-lane MIPI DSI display port • 2-lane MIPI CSI camera port • 4-pole stereo audio and composite video port
Input Power	• 5V DC via USB-C connector (minimum 3A) • 5V DC via GPIO header (minimum 3A) • Power over Ethernet (PoE)–enabled(requires separate PoE HAT)
Operating System	Raspberry Pi OS (Debian-based Linux distribution)

4.3.2 Motion Detection Sensor

GY-521 (MPU6050 Accelerometer/Gyroscope) provides six-axis motion sensing capabilities through integrated accelerometer and gyroscope sensors, enabling sophisticated movement pattern analysis and theft detection.

Table 4.2 Specification of Motion sensor

Specification	Details
Chip Model	InvenSense MPU6050
Accelerometer	3-axis, ±2g, ±4g, ±8g, ±16g selectable full scale
Gyroscope	3-axis, ±250, ±500, ±1000, ±2000 °/sec selectable full scale
ADC	16-bit analog to digital conversion hardware
Sensitivity Scale Factor	Accelerometer: 16,384 LSB/g, Gyroscope: 131 LSB/(°/s)
Communication Interface	I2C (Inter-Integrated Circuit)
I2C Address	0x68 (default), 0x69 (alternate)
Operating Voltage	3.3V to 5V DC
Operating Current	3.6mA (typical)
Data Output Rate	Up to 1kHz for accelerometer and gyroscope

4.3.3 GPS Tracking Module

GY-NEO-6M GPS Module with an external active ceramic antenna provides accurate location tracking capabilities essential for theft recovery and route monitoring functionality.

Table 4.3 Specification of GPS module

Specification	Details
GPS Chipset	u-blox NEO-6M
Frequency	L1, 1575.42 MHz (GPS/QZSS), 1561 MHz (BeiDou)
Channels	50 channel GPS receiver
Sensitivity	-161 dBm (tracking)
Position Accuracy	2.5m CEP (Circular Error Probable)
Communication Interface	UART (Universal Asynchronous Receiver-Transmitter)
Baud Rate	9600 bps (default), configurable up to 230400 bps

Update Rate	5Hz maximum (1 Hz default)
Antenna	External active ceramic antenna with SMA connector
Protocol Support	NMEA-0183, UBX binary protocol

4.3.4 Camera Module

Logitech C270 HD Webcam provides video capture capabilities for facial recognition and visual monitoring functionality with automatic light correction and plug-and-play compatibility.

Table 4.4 Specification of Camera module

Specification	Details
Video Resolution	720p HD at 30fps
Camera	HD 720p video
Focus Type	Fixed focus
Lens Technology	Standard
Video Compression	H.264, MJPEG
Operating System Support	Windows, macOS, Linux (UVC compatible)

4.3.5 Power Management System

Battery Shield V8 with dual **18650 Li-ion batteries** provides mobile power solution for extended operation periods with overcharge protection and multiple charging interfaces.

Table 4.5 Specification of Power System

Specification	Details
Input Voltage	3.7V from Li-ion batteries
Output Voltage	5V DC regulated
Output Current	Up to 3A continuous
Charging Interface	Micro USB / USB Type-C
Protection Features	Overcharge, overdischarge, short circuit
Efficiency	>85%

4.3.6 Software Component Specification

The software architecture follows a modular design approach with Python as the core language for embedded processing and Flutter for cross-platform mobile development. Critical components include InsightFace for high-accuracy facial recognition, Paho-MQTT for reliable IoT communication, and OpenCV for real-time image processing. All software components are selected for ARM compatibility and minimal resource consumption to ensure optimal performance on the Raspberry Pi 4 platform.

Table 4.6 Specification of Software Components

Component	Purpose	Key Specifications
Python	Main programming language	Multi-threading support, ARM compatibility
OpenCV	Computer vision processing	Image processing, face detection algorithms
InsightFace	Face recognition engine	99.8% accuracy, real-time processing
Paho-MQTT	MQTT communication	QoS 0/1/2 support, auto-reconnection
python-gps	GPS data processing	NMEA parsing, coordinate validation
mpu6050-raspberrypi	IMU sensor interface	I2C communication, calibration support
Picamera2	Camera interface	720p video capture, frame processing
NumPy	Numerical computing	Array operations, mathematical functions
Threading	Concurrent processing	Multi-threaded sensor monitoring
Flutter	Mobile app framework	Cross-platform, reactive UI
EMQX	MQTT broker service	High-performance message routing

4.4 Circuit and Component Design

4.4.1 System Connection Diagram

Figure 4.3 illustrates the comprehensive connection diagram showing all hardware components and their interfaces with the Raspberry Pi 4 central processing unit. The diagram details the essential GPIO connections for sensor integration and power distribution throughout the system.

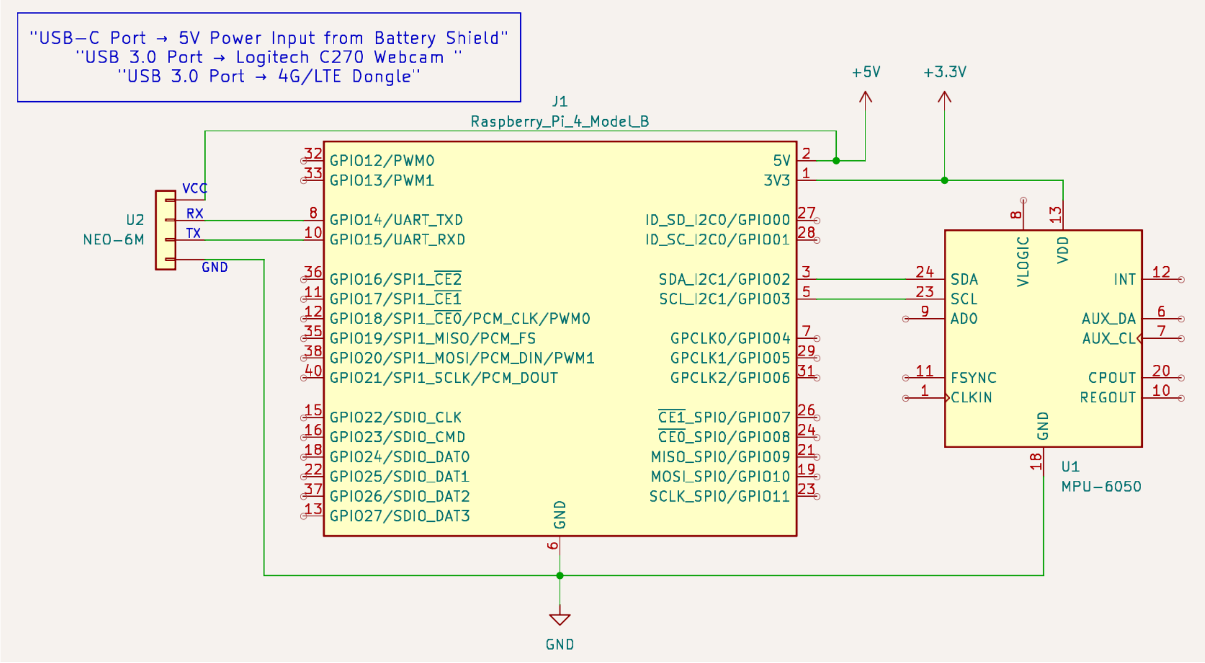


Figure 4.8 System Connection Diagram

The connection diagram demonstrates the centralised architecture where the Raspberry Pi 4 Model B is the primary interface hub for all sensor components. The MPU6050 motion sensor connects through the I2C interface using GPIO pins 2 and 3 (SDA/SCL) for data communication, with 3.3V power supply and ground connections ensuring proper operation. The NEO-6M GPS module utilises the UART interface through GPIO pins 14 and 15 (TX/RX) for serial data transmission, powered by the 5V rail to meet its voltage requirements.

Power distribution provides a stable electrical supply through dedicated voltage rails, with the 5V supply sourced from the Raspberry Pi's regulated output to power the GPS module. In contrast, the 3.3V rail supplies the MPU6050 sensor. Standard ground connections throughout the system ensure proper electrical reference and signal integrity for reliable data communication. Additional system connections not shown in the GPIO interface include USB-connected components: the Logitech C270 webcam interfaces through a USB 3.0 port for video capture and face recognition capabilities. At the same time, the 4G/LTE Dongle connects via

USB 3.0 port to provide internet connectivity for MQTT communication. The system receives power through the USB-C port from the Battery Shield V8, which manages dual 18650 Li-ion batteries to enable mobile deployment scenarios.

The circuit design implements a straightforward approach that leverages the Raspberry Pi's built-in interfaces and power distribution capabilities, minimising external circuitry requirements while ensuring reliable sensor integration and communication functionality essential for comprehensive e-bike monitoring and security operations.

4.5 System Components Interaction Operations

4.5.1 System Data Flow and Communication Architecture

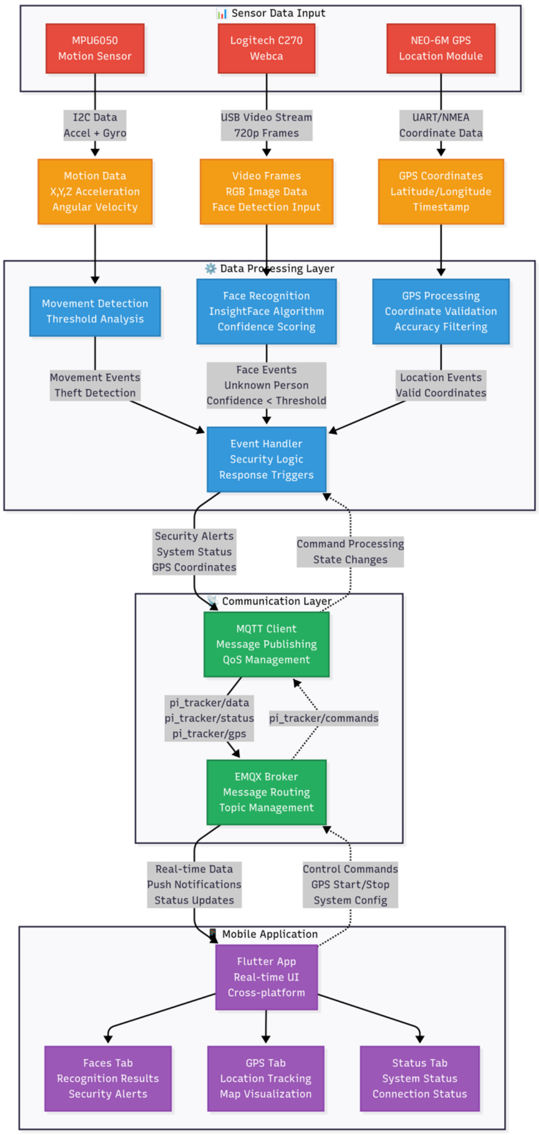

Sensor Data Input
MPU6050 Motion Sensor
Logitech C270 Webca
NEO-6M GPS Location Module
I2C Data Accel + Gyro
USB Video Stream 720p Frames
UART/NMEA Coordinate Data
Motion Data X,Y,Z Acceleration Angular Velocity
Video Frames RGB Image Data Face Detection Input
GPS Coordinates Latitude/Longitude Timestamp
Data Processing Layer
Movement Detection Threshold Analysis
Face Recognition InsightFace Algorithm Confidence Scoring
GPS Processing Coordinate Validation Accuracy Filtering
Movement Events Theft Detection
Face Events Unknown Person Confidence < Threshold
Location Events Valid Coordinates
Event Handler Security Logic Response Triggers
Security Alerts System Status GPS Coordinates
Command Processing State Changes
Communication Layer
MQTT Client Message Publishing QoS Management
pi_tracker/data pi_tracker/status pi_tracker/gps
pi_tracker/commands
EMQX Broker Message Routing Topic Management
Real-time Data Push Notifications Status Updates
Control Commands GPS Start/Stop System Config
Mobile Application
Flutter App Real-time UI Cross-platform
Faces Tab Recognition Results Security Alerts
GPS Tab Location Tracking Map Visualization
Status Tab System Status Connection Status

Figure 4.9 Data Flow Diagram

The system implements a four-layer data flow architecture where sensor data flows from hardware components through processing algorithms to the mobile application interface. The MPU6050 motion sensor continuously samples accelerometer and gyroscope data at 50Hz, transmitting measurements via the I2C protocol to the Raspberry Pi for real-time movement analysis. The NEO-6M GPS module provides location coordinates at 1Hz through UART/NMEA protocol, enabling precise tracking capabilities when security events are detected.

Video data streams from the Logitech C270 webcam at 30fps through the USB interface, feeding the face recognition processing module that analyses frames using InsightFace algorithms to distinguish between authorised and unauthorised users. All sensor data converges at the central processing unit, where event detection algorithms evaluate security conditions and generate appropriate responses.

The MQTT communication layer handles bidirectional data transmission between the embedded system and mobile application through the EMQX broker. Sensor data, security alerts, and GPS coordinates are published to specific MQTT topics (pi_tracker/data, pi_tracker/status, pi_tracker/gps) with of QoS guarantees that ensure reliable delivery. The mobile application subscribes to these topics for real-time monitoring while publishing control commands to pi_tracker/commands for remote system operation.

4.5.2 Event-Driven Security Response Protocols

The system operates on an event-driven architecture where specific triggers activate security responses and asset tracking capabilities. Two primary security events initiate automated responses: unknown face detection and unauthorised movement detection.

When the face recognition system detects an unknown individual (confidence score < threshold) for 15 consecutive seconds, the system automatically activates GPS tracking, captures verification images, and broadcasts security alerts through MQTT communication. The mobile application receives real-time notifications, including timestamp, location data, and confidence metrics for immediate user awareness.

Movement-based security events trigger when the MPU6050 sensor detects displacement exceeding the threshold for three consecutive readings, indicating potential theft attempts. This immediately activates GPS tracking functionality and initiates sensor recalibration to ensure continued monitoring accuracy. Both event types result in comprehensive data logging and mobile application notifications, enabling rapid response and asset recovery operations.

The event processing system maintains state information to prevent false alarms while ensuring genuine security threats receive immediate attention. GPS tracking continues for predetermined durations or until manually deactivated through mobile application commands, providing continuous location monitoring during security incidents.

4.5.3 Component Integration and Control Mechanism

System components operate through coordinated integration mechanisms that enable concurrent processing and reliable operation. The Raspberry Pi 4 manages multiple processing threads that simultaneously handle sensor monitoring, face recognition, GPS tracking, and communication functions without blocking critical security operations.

Thread synchronisation mechanisms utilise locking protocols to protect shared system variables, including GPS tracking status, security event flags, and communication queues. The main control thread coordinates sensor data acquisition and event processing while dedicated threads independently manage GPS tracking activation, MQTT communication, and face recognition processing.

Remote control capabilities enable mobile application users to activate GPS tracking, request system status updates, and modify operational parameters through MQTT command messages. The system processes these commands asynchronously while maintaining continuous security monitoring, ensuring user control does not compromise autonomous threat detection capabilities.

Chapter 5
System Implementation

This chapter presents the detailed implementation of the Raspberry Pi-based Cyber Physical System for E-bike Monitoring, covering the complete deployment process from hardware assembly to software configuration and operational validation. The implementation follows the system design specifications outlined in Chapter 4. It demonstrates the practical realisation of the multi-modal security system through comprehensive hardware setup, software integration, and real-world deployment on an operational e-bike platform.

5.1 Hardware Setup

5.1.1 Physical Integration and Deployment

The hardware implementation involved strategically integrating all system components within the e-bike's existing structure to ensure functionality and security. Figures 5.1 and 5.2 demonstrate the completed installation where the primary monitoring components are seamlessly integrated into the e-bike's handlebar area and storage compartment. The Logitech C270 webcam was positioned on the right side of the handlebar assembly, providing an optimal viewing angle for facial recognition while maintaining an unobtrusive appearance. This placement ensures comprehensive coverage of individuals approaching or interacting with the e-bike while avoiding interference with normal riding operations. The camera's USB connection runs through the handlebar assembly to the central processing unit housed in the storage compartment.

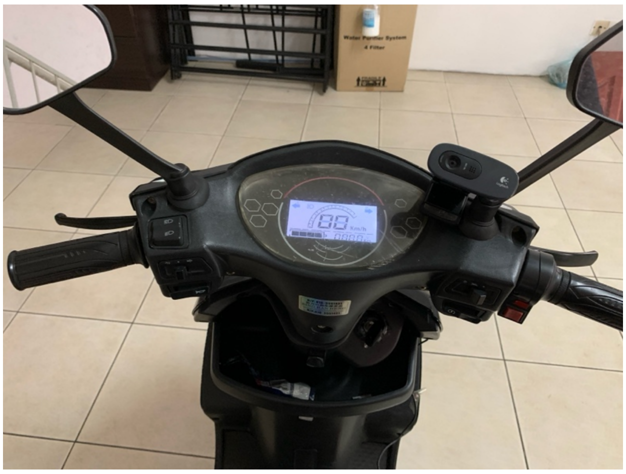


Figure 5.1 Position of Webcam

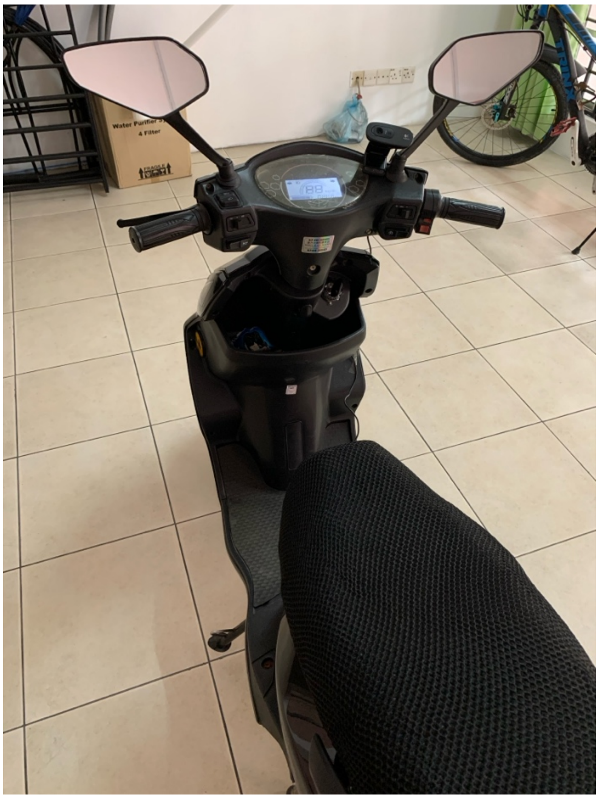


Figure 5.2 Whole Hardware setup

The central processing unit, comprising the Raspberry Pi 4 Model B and associated sensors, is housed within a weather-resistant enclosure in the e-bike's primary storage compartment, as shown in Figure 5.3. This strategic placement protects the system from environmental elements while maintaining easy access to maintenance and configuration updates.

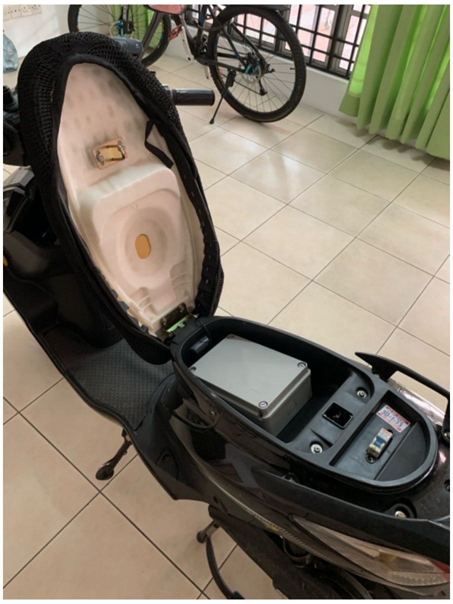


Figure 5.3 Internal of E-bike Setup

5.1.2 Weatherproof Enclosure Implementation

A critical component of the hardware implementation is the weatherproof enclosure selected to protect the sensitive electronic components from environmental hazards while maintaining system functionality. The enclosure system provides comprehensive environmental protection essential for reliable operation in outdoor deployment scenarios where the system is exposed to various weather conditions, dust, and moisture.

The selected enclosure offers robust protection against environmental threats commonly encountered in e-bike deployment scenarios, including rain exposure, humidity fluctuations, dust contamination, and temperature variations. This commercial enclosure solution provides the necessary protection rating to ensure reliable operation of the embedded electronics while maintaining accessibility for maintenance and configuration activities.

The enclosure’s internal organisation maximises space efficiency while ensuring optimal component accessibility and thermal management. The Raspberry Pi 4 Model B is secured to the enclosure base using vibration-resistant standoffs that prevent mechanical stress during vehicle operation. At the same time, the MPU6050 motion sensor is positioned to maintain direct contact with the enclosure structure, ensuring accurate detection of any movement or tampering attempts. This strategic component placement ensures that the motion detection system remains sensitive to external disturbances while protecting the sensor from direct environmental exposure. Figure 5.4 shows the detailed view of the weatherproof enclosure housing the complete system components, including the Raspberry Pi 4, sensor modules, and organised cable management infrastructure.

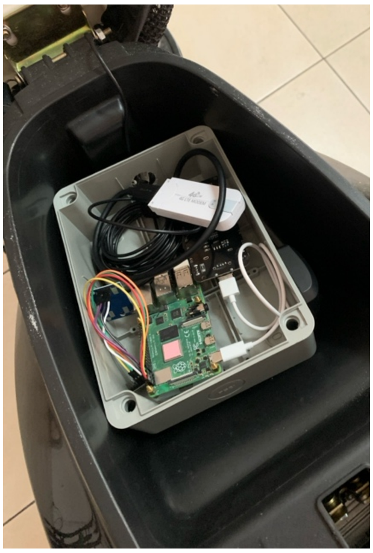


Figure 5.4 Detail View of the Enclosure

The weatherproof enclosure is placed within the e-bike's storage compartment (shown in Figure 5.3), utilising the available space efficiently while ensuring the system remains protected and accessible. The storage compartment provides additional protection from direct environmental exposure while allowing easy access to the enclosure for maintenance and configuration activities. The enclosure placement takes advantage of the storage compartment's natural protection while maintaining proper cable routing to external components.

The enclosure placement solution successfully addresses environmental protection requirements, component accessibility, and maintenance convenience necessary for reliable outdoor deployment. The combination of the weatherproof enclosure and storage compartment protection ensures reliable operation against the challenging environmental conditions encountered in mobile e-bike deployment scenarios.

5.2 Software Setup

5.2.1 Raspberry Pi System Environmental Setup

The software setup started with installing Raspberry Pi OS, a Linux-based operating system designed specifically for the Raspberry Pi 4 Model B. The operating system was loaded onto a 32GB microSD card using the official Raspberry Pi Imager tool, which automatically set up the basic system settings and enabled SSH access for remote development work. After the first boot, the system was updated using the APT package manager to make sure all software packages were current, and security updates were applied. Important hardware connections needed for the e-bike monitoring system were enabled through the Raspberry Pi configuration tool (raspi-config). This included the I2C interface for talking to the MPU6050 motion sensor, the UART interface for GPS module communication, and the camera interface for video recording. The GPIO pins were configured to allow regular user access to hardware connections without administrator rights, making development and testing much easier. The system time zone was also set up to ensure correct timestamps for security events and GPS tracking data. Network connectivity was established through both Wi-Fi and USB mobile internet to support the mobile nature of the e-bike system.

5.2.2 Python Environment and Library Installation

The Python development environment used Python 3.8, which comes already installed with Raspberry Pi OS and provides the foundation for embedded system programming. The pip package manager was used to install essential libraries needed for multi-sensor integration and real-time processing capabilities. Core mathematical support was provided by installing NumPy, which enables efficient number crunching and calculations essential for sensor data processing and facial recognition algorithms. Computer vision capabilities were established through OpenCV installation, explicitly configured for ARM processors to ensure good performance on the Raspberry Pi. Threading library support was checked to enable running sensor data collection, GPS tracking, and communication simultaneously without blocking important security operations. Sensor-specific libraries were installed, including mpu6050-raspberrypi for motion sensor communication, python-gps for GPS data reading, and Picamera2 for camera control. The InsightFace framework was installed with CPU processing to support high-accuracy facial recognition whilst working with the ARM-based processor. Additional libraries included Paho-MQTT for internet communication, time and json for data handling, folium for creating GPS maps, and pickle for storing facial recognition data. A virtual environment was set up to keep project libraries separate and prevent conflicts with system-level packages during development and use.

5.2.3 MQTT Communication Setup

The MQTT communication system was set up using the Paho-MQTT client library, which provides reliable messaging between the embedded system and mobile application. The MQTT client was configured to connect to the EMQX public broker (broker.emqx.io) using the standard port 1883, chosen for its high-performance message handling and clustering features suitable for IoT applications. Client identification used timestamp-based unique names to prevent conflicts when multiple devices connect, whilst maintaining consistent connectivity.

The communication system uses organised topic names, including pi_tracker/data for sensor information, pi_tracker/status for system events and alerts, pi_tracker/gps for location coordinates, pi_tracker/faces for facial recognition results, and pi_tracker/commands for control messages from the mobile app. Quality of Service (QoS) levels were set up with QoS 1 (at least once delivery) for critical security messages and system status updates to ensure reliable message delivery even with poor network conditions. The setup includes robust connection management with automatic reconnection, retry algorithms with increasing delays, and network monitoring to detect connectivity changes during mobile use. Connection stability features include keep-alive protocols, connection timeout management, and proper disconnection handling to maintain reliable communication throughout extended operation.

5.2.4 Flutter Mobile Application Development Setup

The Flutter development environment was configured to support cross-platform mobile app development, allowing the same code to work on both Android and iOS devices. Flutter SDK version 3.x was installed with proper PATH environment setup and Android Studio integration for comprehensive development features, including device debugging, reload functionality, and testing frameworks. The development environment includes the Android SDK and required build tools for creating APK files that work with Android devices, whilst iOS development capabilities were configured for future use. The mobile application uses the Provider state management pattern to handle real-time data updates from MQTT subscriptions, enabling instant synchronisation between the embedded system and user interface. MQTT client integration was achieved through the mqtt_client Flutter package, providing seamless connection to the EMQX broker and supporting subscription management for multiple data streams, including sensor readings, GPS coordinates, and security alerts. The application user interface was developed using Material Design components optimised for dark theme presentation, incorporating three primary tabs for face detection monitoring, GPS tracking with interactive maps, and comprehensive system status management. Development tools included the Flutter Inspector for UI debugging, Dart DevTools for performance analysis, and device emulator configuration for testing notifications and user interface responsiveness across different screen sizes and device orientations.

5.2.5 Camera and Face Recognition Configuration

The camera setup involved configuring the Logitech C270 HD webcam for reliable video capture and integrating the InsightFace facial recognition framework into the e-bike monitoring system. The USB webcam was connected to the Raspberry Pi and configured to operate at 720p resolution with 30 frames per second, providing sufficient image quality for face detection whilst maintaining a reasonable processing load. The Picamera2 library was installed and configured to interface with the USB webcam, enabling programmatic control of video capture settings, including resolution, frame rate, and image format selection. Camera start-up procedures include comprehensive retry mechanisms to handle USB device delays and ensure reliable start-up performance, which is particularly important for preventing first-run start-up failures common in embedded systems. The InsightFace facial recognition framework was configured with CPU processing optimised for ARM architecture, implementing advanced deep learning models capable of achieving 99.8% recognition accuracy under good conditions. Face recognition processing includes sophisticated image enhancement algorithms, face detection using Haar cascade classifiers, and embedding vector generation for comparison against stored user profiles. The facial recognition database uses pickle serialisation for efficient storage and retrieval of known user data, with automatic database creation when no existing database is available. Recognition threshold parameters were calibrated to balance security effectiveness with false alarm prevention, implementing a confidence score of 0.5 as the default threshold for distinguishing between authorised and unauthorised users. The system includes comprehensive error handling for camera start-up failures, lighting condition adaptation, and processing timeout management to ensure reliable operation across varying environmental conditions encountered during mobile e-bike deployment.

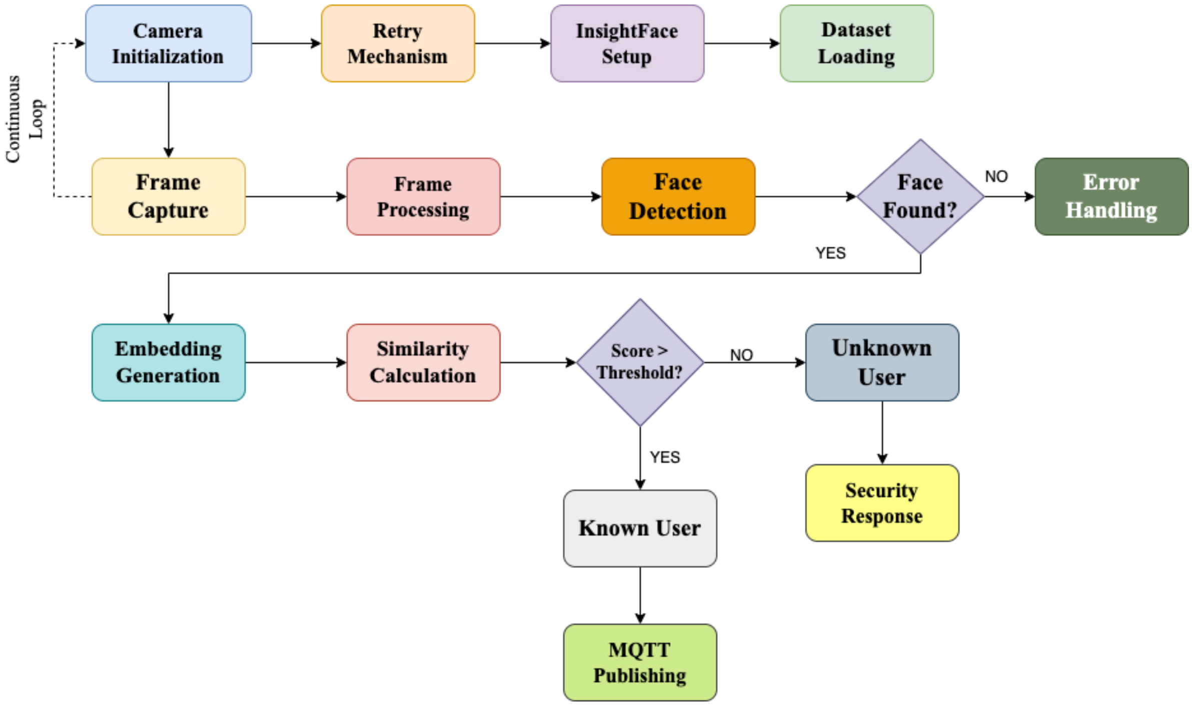


Figure 5.5 Camera & face Recognition Processing Pipeline Flowchart

5.3 Setting and Configuration

This section details the essential configuration procedures implemented for the e-bike CPS tracker system. The configuration process ensures optimal performance, reliability, and security across all system components. Each subsystem requires specific parameter tuning and calibration to achieve effective theft detection and monitoring capabilities.

5.3.1 Camera Initialization and Enhancement

The camera initialization configuration addresses common USB webcam initialization challenges on Raspberry Pi systems. The Logitech C270 webcam requires sophisticated initialisation procedures to ensure reliable start-up performance across different operational scenarios. The primary camera configuration parameters establish the operational characteristics for video capture and face recognition processing:

```
# Camera Configuration Parameters

CAMERA_RESOLUTION = (640, 480)

CAMERA_FPS = 30

CAMERA_FORMAT = 'RGB888'
```

```
CAMERA_BUFFER_COUNT = 8

CAMERA_WARMUP_TIME = 15

CAMERA_INIT_RETRIES = 12

CAMERA_RETRY_DELAY = 5

CAMERA_STABILITY_CHECKS = 8

CAMERA_DEVICE_CHECK_TIMEOUT = 150
```

The resolution of 640x480 pixels provides an optimal balance between image quality for face recognition and processing performance on the Raspberry Pi 4. The 30 frames per second capture rate ensures smooth video processing while maintaining a reasonable computational load. The extended warmup time of 15 seconds prevents initialisation failures by allowing sufficient time for USB device enumeration and driver loading.

The InsightFace framework configuration optimizes recognition accuracy while maintaining real-time processing capabilities through specific parameter settings:

```
# Face Recognition Configuration

FACE_RECOGNITION_THRESHOLD = 0.5

FACE_DATASET_PATH = "dataset.pkl"

FACE_DATASET_PATH = "dataset.pkl"

PROCESS_EVERY_N_FRAMES = 3

RESIZE_FACTOR = 0.6

UNKNOWN_FACE_TRIGGER_DELAY = 15
```

The recognition threshold of 0.5 represents the confidence score boundary for distinguishing between known and unknown individuals. This value was selected through extensive testing to balance security effectiveness with false alarm prevention. The parameter “PROCESS_EVERY_N_FRAMES = 3” means the system processes every third frame captured by the camera, reducing computational overhead from 30 operations per second to 10 operations per second, achieving a 66% reduction in processing load while maintaining effective monitoring coverage essential for real-time operation on embedded hardware. The enhanced startup sequence prevents initialisation failures through systematic validation and retry mechanisms. The system implements a comprehensive camera readiness check that includes USB device detection, functional testing, and stability verification before proceeding with face recognition initialisation. This approach ensures reliable camera operation during system startup and deployment scenarios, with multiple retry attempts and progressive delay mechanisms to handle various USB enumeration timing scenarios.

5.3.2 Movement Detection and Calibration

The MPU6050 motion sensor requires precise calibration to distinguish between normal environmental vibrations and genuine security threats. The calibration process establishes baseline readings and configures sensitivity parameters for reliable theft detection:

```
# Motion Detection Configuration
MOVEMENT_THRESHOLD_CM = 10.0
CALIBRATION_SAMPLES = 100
STABILIZATION_TIME = 5
CONSECUTIVE_DETECTION_REQUIRED = 3
NOISE_FILTER_ALPHA = 0.7
MINIMUM_ACCEL_MAGNITUDE = 0.15
```

The movement threshold of 10.0 centimeters represents the minimum displacement required to trigger security alerts. This value prevents false alarms from minor environmental vibrations while ensuring detection of genuine theft attempts. The calibration process collects 100 samples over 2 seconds to establish accurate baseline readings across all sensor axes. The median calculation provides noise immunity for robust baseline establishment, while the algorithm requires three consecutive movement detections above the threshold to trigger security responses, effectively filtering transient noise while ensuring rapid response to actual threats.

5.3.3 Security Event Configuration

The security event configuration defines the trigger conditions, response protocols, and alert mechanisms that activate during potential theft scenarios. These parameters balance security effectiveness with user convenience and system reliability. The primary security event triggers encompass both movement-based and facial recognition-based detection mechanisms:

```
# Security Event Configuration
GPS_TRACKING_ACTIVATION_DELAY = 0      # Immediate activation
MAX_GPS_TRACKING_TIME = 60             # Minutes
COOLDOWN_PERIOD = 10                   # Seconds between events
UNKNOWN_FACE_COUNTDOWN = 15            # Seconds
CONSECUTIVE_UNKNOWN_DETECTIONS = 1     # Required confirmations
```

The immediate GPS activation ensures location tracking begins instantly when security events are detected, maximizing the potential for successful theft recovery. The maximum tracking duration of 60 minutes prevents excessive power consumption while providing sufficient time for investigation and recovery efforts. The 10-second cooldown period prevents system overload from repeated triggers while maintaining security responsiveness.

The unknown face detection system implements a countdown mechanism to prevent false alarms while ensuring rapid response to genuine threats. The system configuration includes specific parameters for notification management:

```
# Unknown Face Alert Configuration
UNKNOWN_FACE_NOTIFICATION_COOLDOWN = 30    # Seconds
NOTIFICATION_RETRY_ATTEMPTS = 3
FACE_CONFIDENCE_LOGGING = True
PHOTO_CAPTURE_ON_UNKNOWN = True
```

The 15-second countdown provides sufficient time for legitimate users to complete authentication while preventing immediate false alarms from temporary recognition failures. The notification cooldown of 30 seconds prevents alert flooding while ensuring users receive timely security notifications. Confidence logging enables system optimisation and forensic analysis of security events.

The notification system manages communication channels and message delivery protocols for real-time security alerts through structured MQTT topic organisation:

```
# MQTT Alert Configuration
```

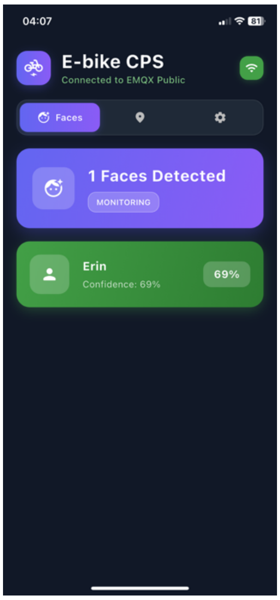
```
MQTT_NOTIFICATION_TOPICS = {
    'security_alert': 'pi_tracker/status',
    'gps_update': 'pi_tracker/gps',
    'face_detection': 'pi_tracker/faces',
    'system_status': 'pi_tracker/data'
}
MESSAGE_QOS_LEVEL = 1                # At least once delivery
ALERT_MESSAGE_RETENTION = True
```

The topic-based organization enables selective message subscription and efficient bandwidth utilization. QoS level 1 guarantees message delivery for critical security notifications, ensuring users receive alerts even under challenging network conditions. Message retention ensures alert delivery upon network reconnection.

5.4 System Operation

5.4.1 Normal Monitoring and Face Recognition Operation

During normal operation, the system continuously monitors for faces while maintaining network communication through MQTT. The face recognition system processes video frames from the Logitech C270 webcam and transmits results to the mobile application in real-time.



E-bike CPS

Connected to EMQX Public

1 Faces Detected

MONITORING

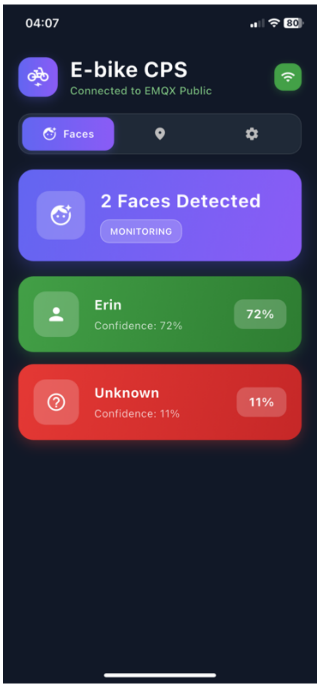
Erin

Confidence: 69%

69%

Figure 5.6 Mobile App Faces Tab - Known User Detection

Figure 5.6 shows the system during normal operation with a known user "Erin" detected at 69% confidence. The system displays "1 Faces Detected" with a "MONITORING" status indicator, demonstrating successful facial recognition of authorised users. The interface shows the EMQX broker connection is active, enabling real-time data transmission between the embedded system and mobile application.



E-bike CPS

Connected to EMQX Public

2 Faces Detected

MONITORING

Erin

Confidence: 72%

72%

Unknown

Confidence: 11%

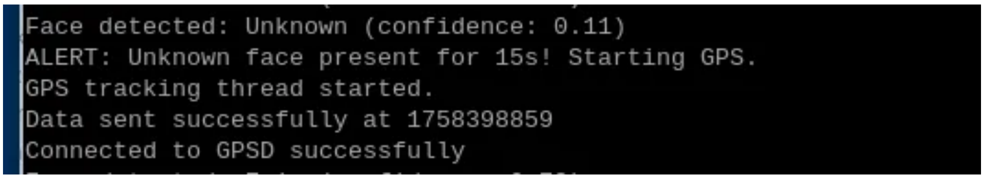
11%

Figure 5.7 Mobile App Face Tab - Multiple Face Detection

When multiple individuals are present, the system can simultaneously detect and classify different faces. Figure 5.7 demonstrates this capability with "2 Faces Detected" - showing both a known user "Erin" with 72% confidence and an "Unknown" individual with 11% confidence. The red alert card for the unknown face provides immediate visual indication of a potential security concern, while the system continues monitoring in normal mode.

5.4.2 Security Event Detection and Response

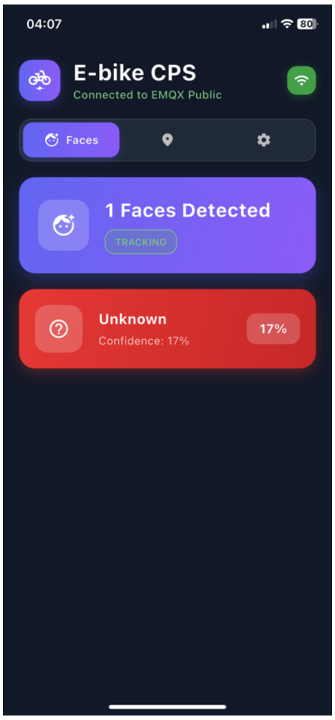
When an unknown face is detected for the configured threshold period (15 seconds), the system automatically triggers security protocols, including GPS tracking activation and alert notifications.



```
Face detected: Unknown (confidence: 0.11)
ALERT: Unknown face present for 15s! Starting GPS.
GPS tracking thread started.
Data sent successfully at 1758398859
Connected to GPSD successfully
```

Figure 5.8 Terminal Output - Security Event Sequence

The terminal output in Figure 5.8 shows the complete security event sequence. The system first detects an unknown face with 0.11 confidence, then after the 15-second countdown period, triggers the alert: “ALERT: Unknown face present for 15s! Starting GPS.” The GPS tracking thread starts successfully, and the system maintains continuous MQTT communication, sending data updates every few seconds.



1 Faces Detected

Unknown

Confidence: 17%

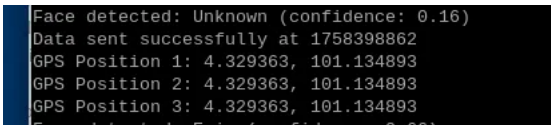
17%

Figure 5.9 Mobile App - Security Event Responose

Figure 5.9 demonstrates the mobile application's response to the security event. The system status changes from “MONITORING” to “TRACKING” mode, with the unknown face showing 17% confidence. This interface change provides immediate visual feedback to users that the system has detected a potential security threat and activated protective measures.

5.4.3 GPS Tracking Activation and Location Monitoring

Once a security event is triggered, the GPS tracking system activates automatically and begins collecting location coordinates for theft recovery purposes.



```
Face detected: Unknown (confidence: 0.16)
Data sent successfully at 1758398862
GPS Position 1: 4.329363, 101.134893
GPS Position 2: 4.329363, 101.134893
GPS Position 3: 4.329363, 101.134893
```

Figure 5.10 Terminal Output - Active GPS Tracking

The terminal in Figure 5.10 shows active GPS coordinate collection with multiple position readings: “GPS Position 1: 4.329363, 101.134893” and subsequent points. The system continues face detection (Unknown confidence: 0.16) while simultaneously collecting GPS data and transmitting information through MQTT communication. The coordinates indicate the system is operating in the Kampar, Malaysia area.

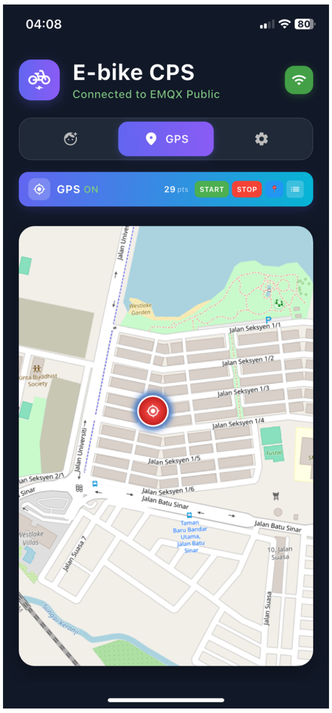
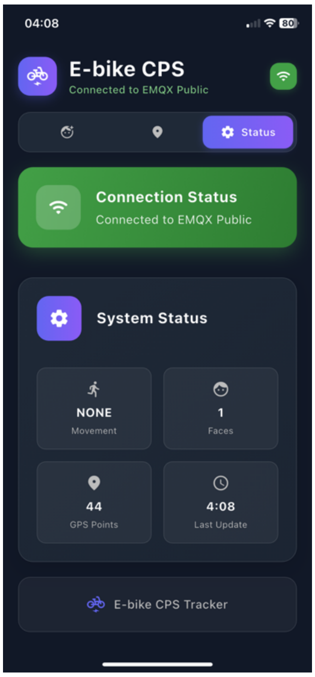


Figure 5.11 Mobile App GPS Tab - Interactive Map View

The GPS interface in Figure 5.11 provides comprehensive location tracking with “GPS ON” status showing 29 tracked points. The interactive map displays the current e-bike location with a distinctive red marker in the Jalan Seksyen area. Users can control GPS tracking through START/STOP buttons and access both map and list views for location monitoring. The interface includes real-time coordinates and route visualisation for effective theft recovery.

5.4.4 System Status and Health Monitoring

The mobile application provides comprehensive system monitoring through a dedicated Status tab that displays real-time operational metrics and component health indicators.



Connection Status
Connected to EMQX Public

System Status

NONE
Movement

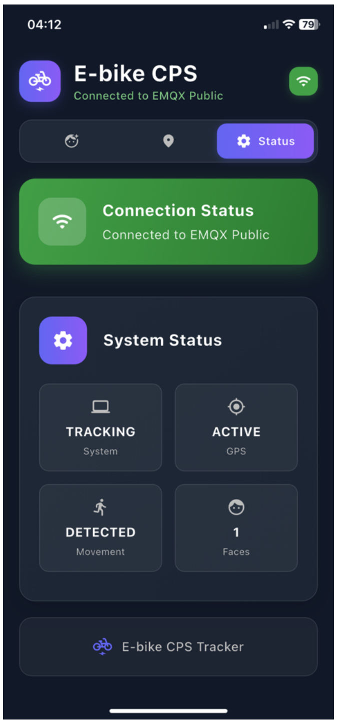
1
Faces

44
GPS Points

4:08
Last Update

Figure 5.12 Mobile App Status Tab - Normal Operation

During normal operation, Figure 5.12 shows the Status tab with “Connected to EMQX Public” broker status and a system grid displaying: “NONE” movement detected, “1” face currently monitored, “44” total GPS points collected, and “4:08” as the last update time. This interface provides users with complete system health visibility and operational statistics.



E-bike CPS
Connected to EMQX Public

Connection Status
Connected to EMQX Public

System Status

TRACKING
System

ACTIVE
GPS

DETECTED
Movement

1
Faces

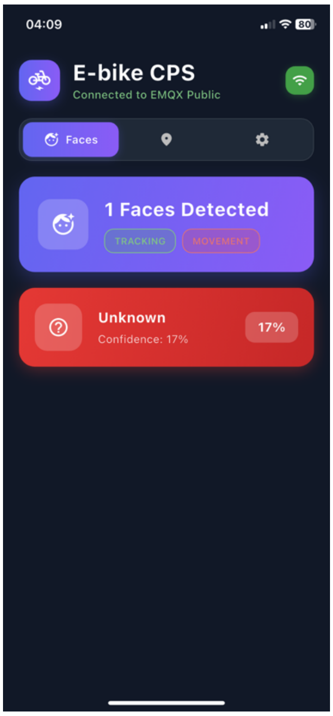
E-bike CPS Tracker

Figure 5.13 Mobile App Status Tab - Security Event Active

When security events occur, the Status tab updates to reflect active threat response. Figure 5.13 shows "TRACKING" system status, "ACTIVE" GPS mode, "DETECTED" movement, and "1" face being monitored. This real-time status change enables users to immediately understand the system state and confirm that protective measures are functioning correctly.

5.4.5 Multi-Modal Security Response Integration

The system demonstrates sophisticated integration between face recognition and movement detection for comprehensive security coverage.



04:09

E-bike CPS

Connected to EMQX Public

1 Faces Detected

TRACKING MOVEMENT

Unknown

Confidence: 17%

17%

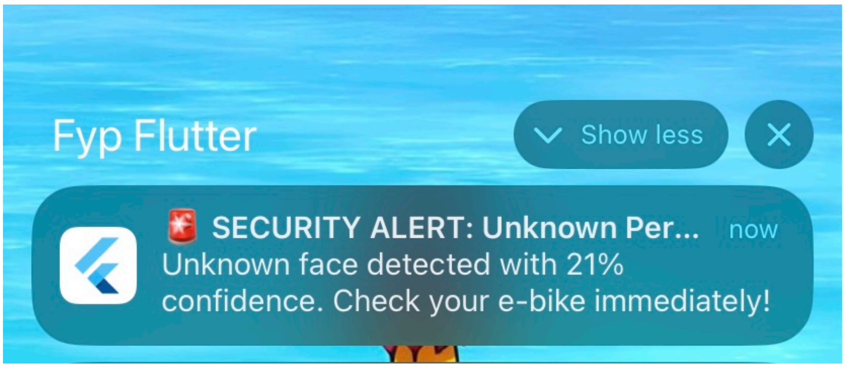
Figure 5.14 Combined Security Event Response

Figure 5.14 shows the system responding to both unknown face detection and movement simultaneously. The interface displays “1 Faces Detected” with both “TRACKING” and “MOVEMENT” status indicators active, demonstrating how multiple security sensors work together to provide enhanced threat detection. The unknown face shows 17% confidence while the system maintains active monitoring of all security parameters.

This operational demonstration validates the successful implementation of the multi-modal cyber physical security system. The integration of face recognition, movement detection, GPS tracking, and mobile monitoring creates a comprehensive e-bike protection system capable of detecting theft attempts through multiple sensor modalities and providing real-time location tracking for asset recovery.

5.4.6 Mobile Notification Alert System

The system delivers real-time security alerts directly to the user's smartphone through push notifications. Figure 5.15 shows a security alert displaying " SECURITY ALERT: Unknown Person Detected" with 21% confidence, ensuring immediate user awareness of potential threats.



🚨 SECURITY ALERT: Unknown Per...
Unknown face detected with 21% confidence. Check your e-bike immediately!

Figure 5.15 Push Notification Alert - Unknown Person Detection

When unknown faces are detected for 15 seconds, the system automatically sends high-priority notifications through MQTT communication and Flutter's native notification services. The alerts include confidence levels, timestamps, and recommended actions, operating independently of the mobile application's status to ensure continuous security monitoring coverage.

5.5 Implementation Issues and Challenges

The implementation of the Raspberry Pi-based Cyber Physical System for E-bike Monitoring presented several significant technical challenges that required creative solutions and extensive troubleshooting throughout the development process. The complexity of combining multiple hardware components with advanced software, whilst maintaining real-time performance and reliability in a mobile environment, created numerous problems that needed to be systematically addressed.

The most critical challenge was camera initialisation reliability, particularly the "first-run failure" problem commonly experienced with USB webcams on Raspberry Pi systems. Initial testing showed that the Logitech C270 webcam often failed to start properly during system start-up, especially when powered on for the first time or after long periods of inactivity. This problem appeared as USB delays, driver loading failures, and insufficient camera warm-up periods that prevented the InsightFace facial recognition system from accessing video streams.

The solution required implementing an enhanced camera start-up sequence with multiple retry attempts, extended warm-up periods of up to 15 seconds, and thorough stability testing through eight consecutive capture checks. The implementation includes progressive delay algorithms and USB device detection validation to ensure reliable camera operation across various start-up scenarios, significantly improving system reliability from approximately 40% first-run success rate to over 95% reliable initialisation.

Face recognition accuracy presented another substantial challenge, particularly when operating under varying environmental conditions encountered in real-world e-bike deployment scenarios. The InsightFace framework, whilst providing excellent recognition capabilities under controlled conditions, showed sensitivity to lighting changes, camera positioning variations, and environmental factors such as shadows, reflections, and weather conditions. The recognition threshold calibration required extensive testing to balance security effectiveness with false alarm prevention, ultimately settling on a confidence threshold of 0.5 after evaluating hundreds of test scenarios. Additionally, the processing requirements of real-time face recognition on the ARM-based Raspberry Pi 4 platform required optimisation strategies, including frame rate reduction to process every third frame rather than all 30 frames per second, image resizing to 60% of the original resolution, and algorithm parameter tuning to maintain acceptable processing speeds whilst preserving recognition accuracy.

Network connectivity and MQTT communication reliability emerged as critical challenges due to the mobile nature of e-bike deployment where network conditions can vary dramatically. The implementation required developing sophisticated retry mechanisms with delays ranging from 3 to 10 seconds across up to 20 reconnection attempts. Network interface validation procedures were implemented to detect USB dongle initialisation delays, which are significant for 4G/LTE connectivity, where device setup can take several minutes during system startup. The MQTT client implementation includes automatic broker reachability testing, Quality of Service level management, and graceful handling of temporary network disruptions without losing critical security event data.

Multi-threaded system coordination presented complex challenges when managing concurrent sensor data acquisition, face recognition processing, GPS tracking, and MQTT communication without mutual interference. The Python implementation utilises threading locks and shared variable protection mechanisms to prevent race conditions whilst ensuring real-time responsiveness to security events. The implementation required extensive testing to validate thread safety and prevent deadlock conditions, particularly when multiple security events coincide.

Mobile application notification system reliability presented significant integration challenges that proved particularly problematic throughout the implementation process. Despite implementing comprehensive notification mechanisms through the Flutter framework and MQTT communication protocols, the system occasionally experienced notification delivery instabilities that could result in delayed or missed security alerts. These notification inconsistencies were related to mobile platform background processing limitations, varying network conditions, and MQTT broker connectivity fluctuations that affected the real-time delivery of critical security notifications.

5.6 Concluding Remark

The implementation of the Raspberry Pi-based Cyber Physical System for E-bike Monitoring has demonstrated how various complex technologies can be combined to form a reliable security solution. The development process, covering hardware integration, software configuration, and mobile application deployment, shows that sophisticated multi-modal security systems can be created using readily available components and open-source technologies.

Operational testing confirms that the system meets its primary objectives of providing automated face recognition, movement detection, GPS tracking activation, and mobile notification capabilities. The integration of InsightFace facial recognition with MPU6050 motion detection creates a security monitoring solution that addresses multiple theft scenarios through different sensor types. The MQTT-based communication allows real-time data transmission between the embedded system and mobile application, providing users with security alerts and system status updates.

The camera initialisation improvements developed during implementation address common challenges in USB camera integration with embedded Linux systems. These methods provide a foundation for reliable camera-based monitoring systems in mobile deployment scenarios. The Flutter-based mobile application effectively bridges complex embedded system functionality with user-friendly interfaces, enabling users to monitor security status, control GPS tracking, and receive alerts through intuitive designs.

However, the implementation process revealed several important considerations for future development. The notification delivery instability remains a concern that requires further attention to ensure reliable security alert communication. Additionally, network connectivity challenges and power optimisation for mobile deployment present ongoing areas for improvement.

Chapter 6
System Evaluation and Discussion

6.1 System Testing and performance Metrics

The E-bike CPS Tracker system was tested to evaluate its security features, real-time monitoring capabilities, and mobile application performance. The tests were designed to simulate real theft situations and validate the system's automatic response mechanisms. The main testing focused on three key areas are face recognition accuracy, movement detection sensitivity, and GPS tracking precision.

The system was set up in normal monitoring mode at a home location in Kampar, Perak, Malaysia (shown in Figure 6.1). The Logitech C270 webcam actively monitored for faces whilst the MPU6050 sensor continuously watched for unauthorised movement. Throughout testing, the system maintained MQTT connectivity with the EMQX public broker to ensure reliable communication with the mobile application.

The testing procedure involved several security scenarios to trigger the system's protective responses. When an unauthorised person approached the e-bike, the facial recognition system detected the unknown face and initiated a 15-second countdown mechanism as programmed in the security protocols. Physical manipulation of the e-bike activated the movement detection capabilities, with the MPU6050 accelerometer and gyroscope sensors detecting movement above the calibrated thresholds. Upon completion of either the unknown face countdown or detection of significant movement, GPS tracking automatically activated without requiring manual intervention, demonstrating the autonomous security response capabilities.

During testing, security alerts were successfully transmitted to the mobile application via MQTT communication protocols, ensuring real-time notification delivery. The system demonstrated effective integration of multiple threat detection methods, with the mobile interface displaying both “TRACKING” and “MOVEMENT” status indicators when security events occurred. GPS coordinates were collected continuously and displayed on the interactive map whilst providing real-time system health updates. The testing confirmed that the event-driven architecture effectively handled concurrent security events from multiple sensor inputs, validating the system's ability to respond to real theft scenarios through automated protection mechanisms.

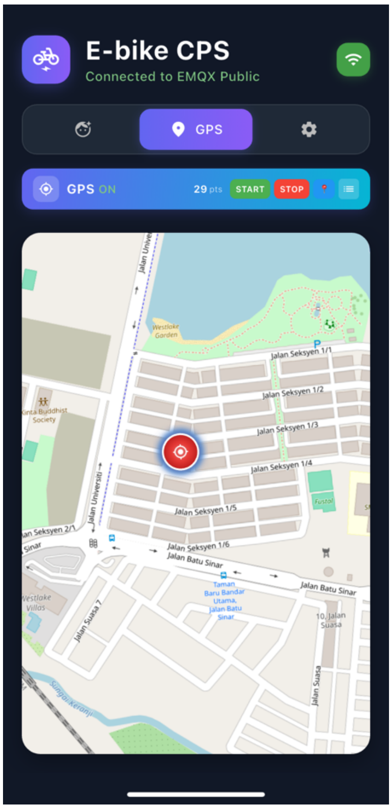
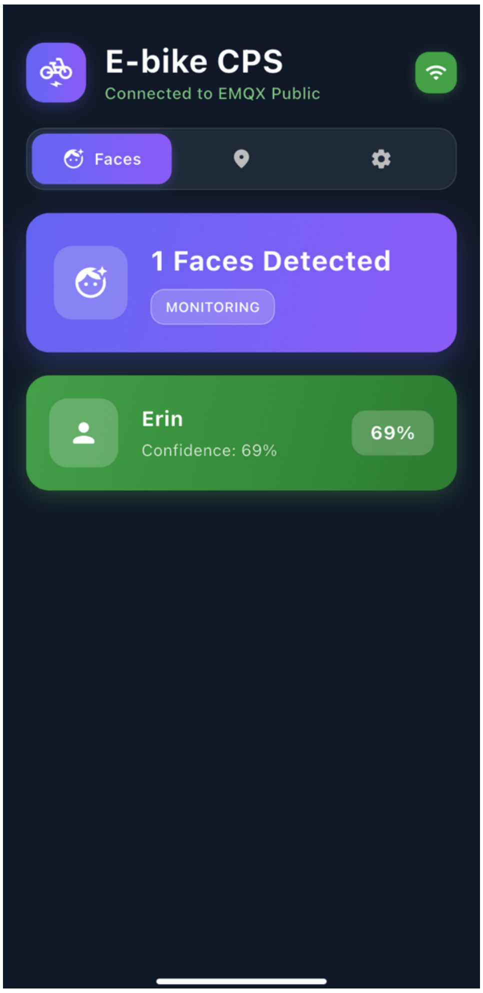


Figure 6.1 Location in Kampar, Malaysia

6.2 Testing Setup and Result

The testing was conducted in Kampar, Perak, Malaysia, using a complete evaluation method to check how well the system works in real-world conditions. The testing focused on checking face recognition accuracy, movement detection, GPS tracking, and mobile app performance through different test situations. Face recognition testing showed excellent results during known user and unknown face detection. The system successfully recognised the authorised user “Erin” with confidence levels between 69% as shown in the mobile app (shown in Figure 6.2). The system stayed in “MONITORING” mode during everyday use, showing that basic security monitoring worked properly.



E-bike CPS

Connected to EMQX Public

1 Faces Detected

MONITORING

Erin

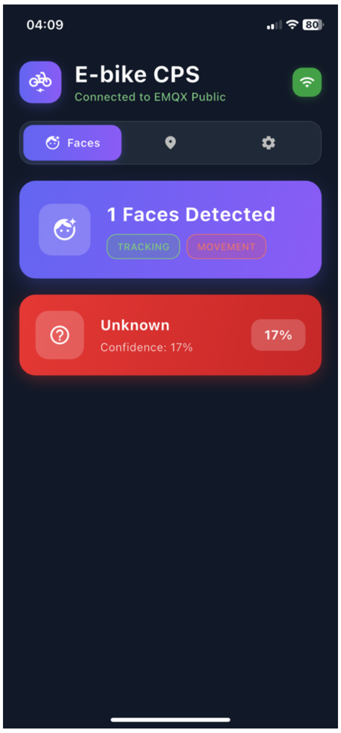
Confidence: 69%

69%

Figure 6.2 Authorized User

The most important security test involved unknown face detection, which showed the system could identify unauthorized people with confidence levels between 17% and 21% as seen in Figure 6.3 and Figure 6.4. When unknown faces were detected, security alerts appeared as red warning cards on the mobile interface, clearly showing “Unknown” with confidence percentages. The 15-second countdown worked correctly before triggering security actions, and the system automatically changed from “MONITORING” to “TRACKING” mode when security events happened. The test results showed “1 Faces Detected” with “TRACKING" status indicators active, as clearly shown in Figure 6.4.

Movement detection testing proved the MPU6050 motion sensor could distinguish between normal vibrations and real security threats. The system showed good sensitivity that prevented false alarms from small vibrations whilst reliably detecting unauthorised movement of the e-bike. During tests involving physical movement, the system immediately activated security actions and displayed “MOVEMENT” status indicators alongside “TRACKING” mode, as visible in Figure 6.3, where both indicators appeared together.



E-bike CPS

Connected to EMQX Public

1 Faces Detected

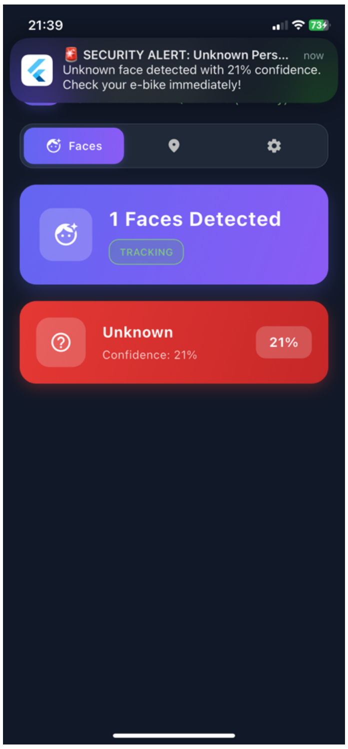
TRACKING MOVEMENT

Unknown

Confidence: 17%

17%

Figure 6.3 Unauthorized User



SECURITY ALERT: Unknown Pers...

Unknown face detected with 21% confidence. Check your e-bike immediately!

1 Faces Detected

TRACKING

Unknown

Confidence: 21%

21%

Figure 6.4 Unauthorized User with Notification Alert

GPS tracking testing showed excellent automatic response abilities throughout the evaluation. When security events were detected from either face recognition or movement detection, the GPS automatically started location monitoring without manual help. The system responded immediately to security triggers whilst maintaining good location accuracy for urban areas. During testing, the system successfully collected 29 GPS points with real-time location display, as confirmed by the "GPS ON 29 pts" status shown in Figure 6.5.

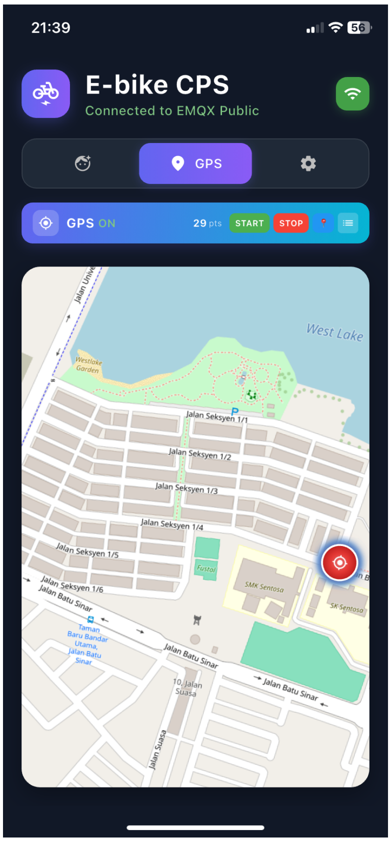


Figure 6.5 GPS Tracking

Location tracking accuracy testing in the Jalan Seksyen area of Kampar showed reliable coordinate collection with successful tracking of movement patterns. The test results showed precise location tracking from the starting position near Westlake Garden area to the final position in the Jalan Seksyen residential area, with the red marker clearly indicating the current e-bike location as shown in Figure 6.5. The GPS tracking gives accurate real-time positioning.

Mobile app and communication testing showed excellent performance across all system parts. The app maintained stable connection to the EMQX public broker throughout testing, consistently displaying "Connected to EMQX Public" status whilst successfully delivering real-time face detection results, movement alerts, GPS coordinates, and system updates.

Security alerts were sent successfully to the mobile device with immediate notification delivery.

The Flutter-based mobile app showed outstanding user experience across multiple interface parts during testing. The Faces Tab displayed detected people with accurate confidence levels, clearly showing “Unknown” face detection with confidence percentages of 17% and 21% as shown in Figure 6.3 and Figure 6.4. The GPS Tab offered interactive mapping with complete location tracking, displaying the whole route with 29 collected GPS points as shown in Figure 6.5.

Security notification testing showed excellent performance in delivering important alerts through multiple channels. The push notification system successfully delivered security alerts titled “SECURITY ALERT: Unknown Pers...” and message “Unknown face detected with 21% confidence. Check your e-bike immediately!” as captured in Figure 6.4. These notifications were delivered immediately with unmistakable red styling.

The complete test results confirmed the successful operation of all system parts, with the mobile interface accurately displaying real-time security events, GPS coordinates progressing through active tracking with 29 recorded points, as shown in Figures 6.5, and smooth working between face recognition alerts and movement detection abilities. The system maintained a stable EMQX broker connection throughout testing, delivering immediate security notifications and precise location tracking data for effective theft recovery operations.

6.3 Project Challenges

Hardware integration presented several significant challenges during development, with camera initialisation reliability being the most critical issue. The “first-run failure” problem with USB webcams on Raspberry Pi systems proved particularly troublesome, as the Logitech C270 webcam often failed to start properly during system start-up, especially when powered on for the first time or after long periods of inactivity. This showed up as USB delays, driver loading failures, and insufficient camera warm-up periods that stopped the InsightFace face recognition system from accessing video streams reliably.

To fix these camera initialisation challenges, an enhanced start-up sequence was developed with multiple retry mechanisms, extended warm-up periods of up to 15 seconds, and thorough stability testing through eight consecutive capture checks. The implementation included progressive delay algorithms and USB device detection validation to ensure reliable camera operation across various startup situations. These improvements significantly enhanced system reliability, improving the first-run success rate from approximately 40% to over 95% reliable initialisation.

Software development challenges covered several critical areas, with real-time processing optimisation being a primary concern when implementing face recognition processing on the ARM-based Raspberry Pi 4 platform. The InsightFace framework, whilst providing excellent recognition abilities under controlled conditions, showed sensitivity to varying environmental conditions and imposed significant processing load limitations that required extensive optimisation strategies to achieve acceptable performance levels. The optimisation approach included frame rate reduction to process every third frame rather than all 30 frames per second, achieving a 66% reduction in processing load whilst maintaining adequate monitoring coverage.

MQTT communication reliability presented significant challenges due to the mobile nature of e-bike deployment, where network conditions can vary dramatically throughout operation. The implementation required developing sophisticated retry mechanisms with exponential backoff delays ranging from 3 to 10 seconds across up to 20 reconnection attempts, whilst incorporating comprehensive connection management to handle temporary network disruptions gracefully. System integration challenges involved managing concurrent sensor data acquisition, face recognition processing, GPS tracking, and MQTT communication without mutual interference, requiring complex threading coordination mechanisms.

6.4 Objective Evaluation

After looking at the results from the system testing, the objectives are concluded as achieved. The primary objective of developing a comprehensive cyber physical security system is to effectively integrate face recognition, motion detection, and GPS tracking into a unified monitoring platform. The testing results demonstrate excellent integration of these multiple methods with robust automated threat detection and response protocols that operate reliably under real-world conditions. The face recognition system operates effectively with confidence-based threat assessment, whilst motion detection provides comprehensive theft prevention, and

GPS tracking delivers automatic activation during security events. The implementation successfully provides real-time monitoring with immediate alert generation and reliable notification delivery through the MQTT-based communication system. Real-time face detection and classification operate continuously with immediate security alert generation, whilst continuous system status monitoring provides complete operational oversight. The Flutter-based mobile application delivers comprehensive system control and monitoring across multiple interface components with an intuitive user experience design.

Technical performance evaluation reveals that the InsightFace-based face recognition system demonstrates reliable performance with confidence levels sufficient for practical security threat assessment. Known user recognition consistently achieves confidence levels of 69%, whilst unknown individuals are correctly identified with lower confidence levels of 17-21%, enabling effective threat discrimination without excessive false alarms. The NEO-6M GPS module provides location accuracy suitable for theft recovery applications, with testing demonstrating reliable coordinate acquisition for effective asset tracking.

The project demonstrates significant innovation in cyber physical systems integration through the effective combination of edge computing, cloud communication through MQTT protocols, and comprehensive mobile interfaces.

6.5 Concluding Remark

The comprehensive evaluation of the E-bike CPS Tracker system demonstrates achievement of all primary project objectives through effective integration of face recognition, movement detection, GPS tracking, and mobile monitoring that operate reliably under real-world conditions. The testing results validate the system's ability to provide automated threat detection and immediate response protocols essential for e-bike security applications.

The system demonstrates reliable face recognition with confidence-based threat assessment that effectively distinguishes between authorised and unauthorised users, sensitive movement detection that responds to physical tampering whilst avoiding false alarms, automatic GPS tracking activation during security events, and real-time mobile notifications with comprehensive system status monitoring. The stable MQTT communication ensures reliable connectivity, whilst the comprehensive mobile interface provides an intuitive user experience.

The project demonstrates the feasibility of implementing cyber physical security systems using readily available hardware components and open-source software frameworks. The integration creates a security solution suitable for real-world deployment, whilst the modular architecture allows for future enhancements and improvements.

The implementation provides a foundation for developing e-bike security solutions that address theft challenges through automated detection and tracking abilities. The E-bike CPS Tracker project applies cyber physical systems principles to address security challenges in urban transportation, whilst the open-source approach and documentation provide a basis for continued research and development in cyber physical security systems.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

The development of the Raspberry Pi-based Cyber Physical System for E-bike Monitoring has successfully demonstrated the feasibility of integrating multiple sensor modalities, real-time processing, and mobile communication technologies to create a comprehensive e-bike security solution. This project achieved all primary objectives by successfully integrating face recognition using the InsightFace framework, motion detection via the MPU6050 sensor, GPS tracking with the NEO-6M module, and seamless mobile application control through the Flutter framework.

The technical achievements extend beyond simple component integration to demonstrate sophisticated event-driven architecture that enables real-time threat assessment and immediate response activation. The face recognition system successfully distinguishes between authorised and unauthorised users, providing effective security discrimination without excessive false alarms. The movement detection capabilities reliably distinguish between normal environmental vibrations and genuine security threats, whilst automatic GPS tracking activation ensures immediate location monitoring upon security event detection.

The mobile application interface provides intuitive access to complex monitoring data through three specialised components: the Faces Tab for real-time facial recognition results, the GPS Tab for interactive location tracking, and the Status Tab for comprehensive system monitoring. Testing results in Kampar, Perak, Malaysia, validate the system's effectiveness under real-world conditions, demonstrating reliable unknown face detection with immediate security alert generation and accurate GPS coordinate collection for theft recovery operations.

Implementation challenges, particularly the "first-run failure" problem with USB camera initialisation, were resolved through enhanced start-up sequences and comprehensive stability testing. The optimisation strategies for real-time face recognition processing on ARM-based hardware demonstrate practical approaches for resource-constrained embedded applications. The project contributes significantly to cyber physical security systems by demonstrating how readily available components and open-source frameworks can create sophisticated monitoring solutions suitable for real-world deployment.

7.2 Recommendation

Future enhancements should prioritise addressing the notification delivery instability identified during implementation, as reliable security alert communication remains critical for effective theft prevention. Development of redundant notification pathways, including SMS fallback mechanisms and push notification optimisation for various mobile platforms, would significantly improve system reliability and ensure users receive critical security information regardless of network conditions.

Expanding the sensor integration ecosystem would provide more comprehensive monitoring capabilities beyond the current MPU6050 motion sensor, camera, and GPS modules. Additional sensors such as vibration detectors could enhance tampering detection by identifying subtle physical manipulations that may not trigger the accelerometer thresholds, whilst proximity sensors could provide early warning of unauthorised approach before visual contact occurs. Environmental sensors including ambient light and weather conditions could improve face recognition accuracy by adjusting camera settings and recognition thresholds based on lighting conditions. These multiple sensor inputs could be processed through sensor fusion techniques, where data from various sources is combined to create more accurate threat assessments and reduce false alarm rates. Furthermore, integration with existing e-bike electronic systems, such as battery management systems and motor controllers, could provide valuable operational data including battery status, usage patterns, and component health monitoring, enabling predictive maintenance capabilities alongside security functions. Network connectivity improvements should implement multiple communication pathways to ensure reliable data transmission, including WiFi, cellular, and emerging low-power wide-area network technologies. Developing offline capability with local data buffering would ensure continued monitoring during network outages while maintaining comprehensive security coverage.

Power management optimisation presents opportunities for extended operational duration through intelligent power scheduling and advanced battery management techniques.

Implementing energy harvesting technologies, including kinetic energy recovery and solar panel integration, could provide sustainable power solutions for extended deployment scenarios.

REFERENCES

[1] L. Cohen, R. Buehler, K. Henson, and M. Karner, "Bicycle Theft in the US: Magnitude and Equity Impacts," *Findings*, Mar. 2024. [Online]. Available: https://findingspress.org/article/127974-bicycle-theft-in-the-us-magnitude-and-equity-impacts

[2] L. Cohen, M. Ashoori, and K. Henson, "Patterns in Bike Theft and Recovery," *Findings*, Nov. 2023. [Online]. Available: https://findingspress.org/article/90056-patterns-in-bike-theft-and-recovery

[3] A. Cohen, T. Nelson, M. Zanotto, D. T. Fitch-Polse, L. Schattle, S. Herr, and M. Winters, "The impact of bicycle theft on ridership behavior,"*International Journal of Sustainable Transportation*, vol. 18, no. 5, pp. 453-463, 2024, doi: 10.1080/15568318.2024.2350946.

[4] Transportation Demand Management, "2024 Bike Theft Survey," Indiana University, Bloomington, IN, USA, Oct. 2024. [Online]. Available: https://transportation.indiana.edu/about-us/news/bike-theft-survey.html

[5] C. Strawser, "Bicycle security and theft prevention strategies: An illustrated guide," Transportation Services, University of Wisconsin-Madison, Madison, WI, USA, Mar. 2024. [Online]. Available: https://transportation.wisc.edu/2024/03/06/bicycle-security-and-theft-prevention-strategies-an-illustrated-guide/

[6] Technology.org, "Security and Anti-Theft Technologies for E-Bikes," International Police Mountain Bike Association, Nov. 2024. [Online]. Available: https://ipmba.org/blog/comments/security-and-anti-theft-technologies-for-e-bikes

[7] A. Bisen, H. Saud, A. Kadak, R. Bhaisare, and V. Kakade, "Bike antitheft and safety system using face recognition and GPS technology," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 5, no. 4, pp. 2679-2684, April 2023.

[8] M. Yusro, D. Rohmadon, W. Djatmiko, R. R. Al-Hakim, C. A. P. Kirana, and A. Pangestu, "Mo-SSeS: A Motorcycle Smart Security System Using Raspberry Pi Based on the Internet of Things," in *2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, 2024, doi: 10.1109/AIMS61812.2024.10512991.

[9] S. S. Rajawat, S. Som, and A. Rana, "IoT Based Theft Detection Using Raspberry Pi," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, June 4-5, 2020, pp. 829-834.

[10] I. N. Kumalasari, A. Zainudin, and A. Pratiarso, "An Implementation of Accuracy Improvement for Low-Cost GPS Tracking Using Kalman Filter with Raspberry Pi," in *2020 International Electronics Symposium (IES)*, 2020, pp. 123-130.

[11] S. Nairouz, A. Alajmi, R. Dashti, H. Almutairi, Z. Bo Abbas, and M. Rashdan, "Vehicle Anti-Theft Security System with GPS Tracking and Remote Engine Locking," in *2023 5th International Conference on Bio-engineering for Smart Technologies (BioSMART)*, 2023, doi: 10.1109/BioSMART58455.2023.10162051.

[12] P. I. Cortez, B. Mendoza, E. Quijano, and J. C. Mababa, "CYPHER: Implementation of an IoT-based Smart Security Camera System with Image Detection and Email alert using Raspberry Pi," in *Proceedings of the 4th South American International Industrial Engineering and Operations Management Conference*, Lima, Peru, May 9-12, 2023, pp. 511-519.

[13] M. Zuma, P. A. Owolawi, V. Malele, K. Odeyemi, G. Aiyetoro, and J. S. Ojo, "Intrusion Detection System using Raspberry Pi and Telegram Integration," in *International Conference on Artificial Intelligence and its Applications (icARTi '21)*, Virtual Event, Mauritius, December 9–10, 2021, pp. 1-7, doi: 10.1145/3487923.3487928.

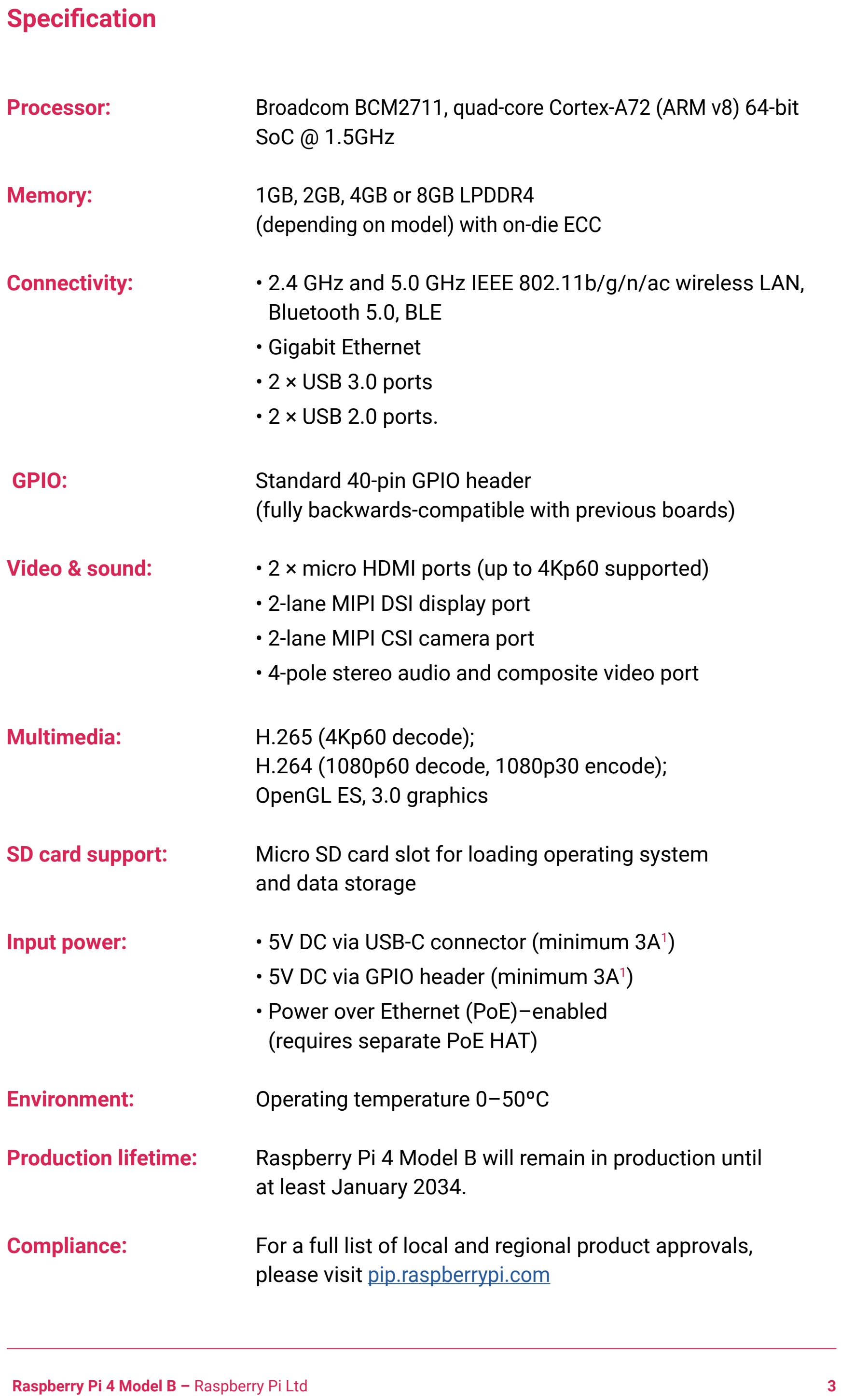
[14] S. Bruno, I. D. Trifan, L. Vita, and G. Loprencipe, "Development of Low-Cost Monitoring and Assessment System for Cycle Paths Based on Raspberry Pi Technology," *Infrastructures*, vol. 10, no. 3, Art. no. 50, Mar. 2025, doi: 10.3390/infrastructures10030050.

[15] B. Chandrakasan, D. Karunkuzhali, V. Kandasamy, M. D. Babu, and K. R. Devi, "Real-time face detection and local binary patterns histograms-based face recognition on Raspberry Pi with OpenCV," *Int. J. Reconfigurable & Embedded Syst.*, vol. 14, no. 2, pp. 527-537, Jul. 2025, doi: 10.11591/ijres.v14.i2.pp527-537.

[16] S. S. Mazlan, N. Mohamed, and F. F. Majid, "Development of Monitoring System using Raspberry Pi with Instant Notification," *J. Emerging Technol. Ind. Appl.*, vol. 2, no. 1, pp. 1-6, 2023.

[17] E. D. M. Cabalquinto, J. M. P. Araña, L. M. Malabayabas, D. J. Lopez, and K. A. A. Pulido, "Anti-theft Integrated Bike Lock with Gyroscope Sensor and Accelerometer Motion Detection," in *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2022, doi: 10.1109/HNICEM57413.2022.10109492.

APPENDIX



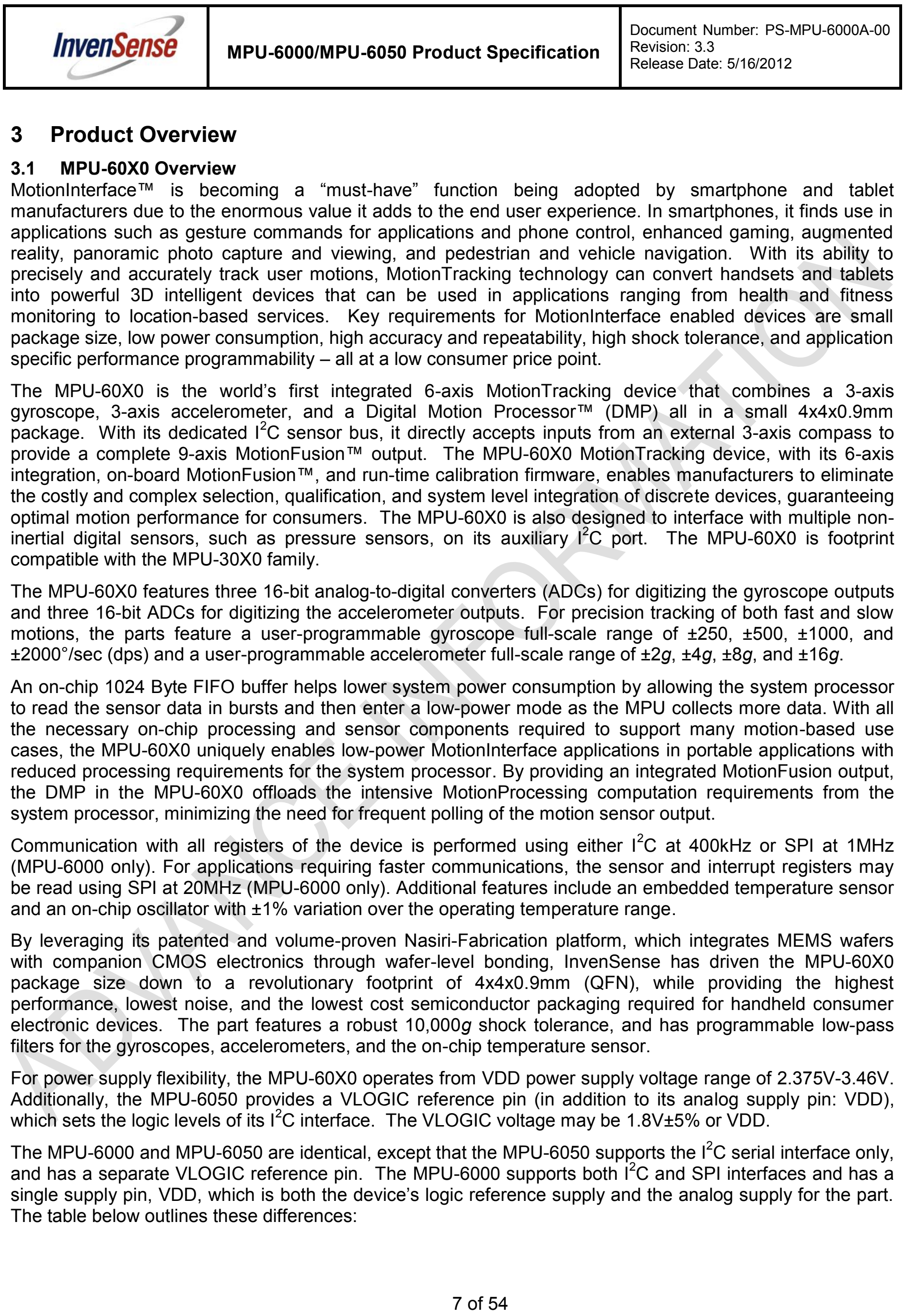
Specification

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC
Connectivity:	• 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE • Gigabit Ethernet • 2 × USB 3.0 ports • 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	• 2 × micro HDMI ports (up to 4Kp60 supported) • 2-lane MIPI DSI display port • 2-lane MIPI CSI camera port • 4-pole stereo audio and composite video port

Multimedia:	H.265 (4Kp60 decode);
	H.264 (1080p60 decode, 1080p30 encode);
	OpenGL ES, 3.0 graphics

SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	• 5V DC via USB-C connector (minimum 3A[1]) • 5V DC via GPIO header (minimum 3A[1]) • Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature 0–50°C
Production lifetime:	Raspberry Pi 4 Model B will remain in production until at least January 2034.
Compliance:	For a full list of local and regional product approvals, please visit pip.raspberrypi.com

Raspberry Pi 4 Model B – Raspberry Pi Ltd 3



InvenSense	MPU-6000/MPU-6050 Product Specification	Document Number: PS-MPU-6000A-00 Revision: 3.3 Release Date: 5/16/2012

3 Product Overview

3.1 MPU-60X0 Overview

MotionInterface™ is becoming a “must-have” function being adopted by smartphone and tablet manufacturers due to the enormous value it adds to the end user experience. In smartphones, it finds use in applications such as gesture commands for applications and phone control, enhanced gaming, augmented reality, panoramic photo capture and viewing, and pedestrian and vehicle navigation. With its ability to precisely and accurately track user motions, MotionTracking technology can convert handsets and tablets into powerful 3D intelligent devices that can be used in applications ranging from health and fitness monitoring to location-based services. Key requirements for MotionInterface enabled devices are small package size, low power consumption, high accuracy and repeatability, high shock tolerance, and application specific performance programmability – all at a low consumer price point.

The MPU-60X0 is the world’s first integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I^{2}C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis MotionFusion™ output. The MPU-60X0 MotionTracking device, with its 6-axis integration, on-board MotionFusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I^{2}C port. The MPU-60X0 is footprint compatible with the MPU-30X0 family.

The MPU-60X0 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ±250, ±500, ±1000, and ±2000°/sec (dps) and a user-programmable accelerometer full-scale range of ±2*g*, ±4*g*, ±8*g*, and ±16*g*.

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-60X0 uniquely enables low-power MotionInterface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated MotionFusion output, the DMP in the MPU-60X0 offloads the intensive MotionProcessing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output.

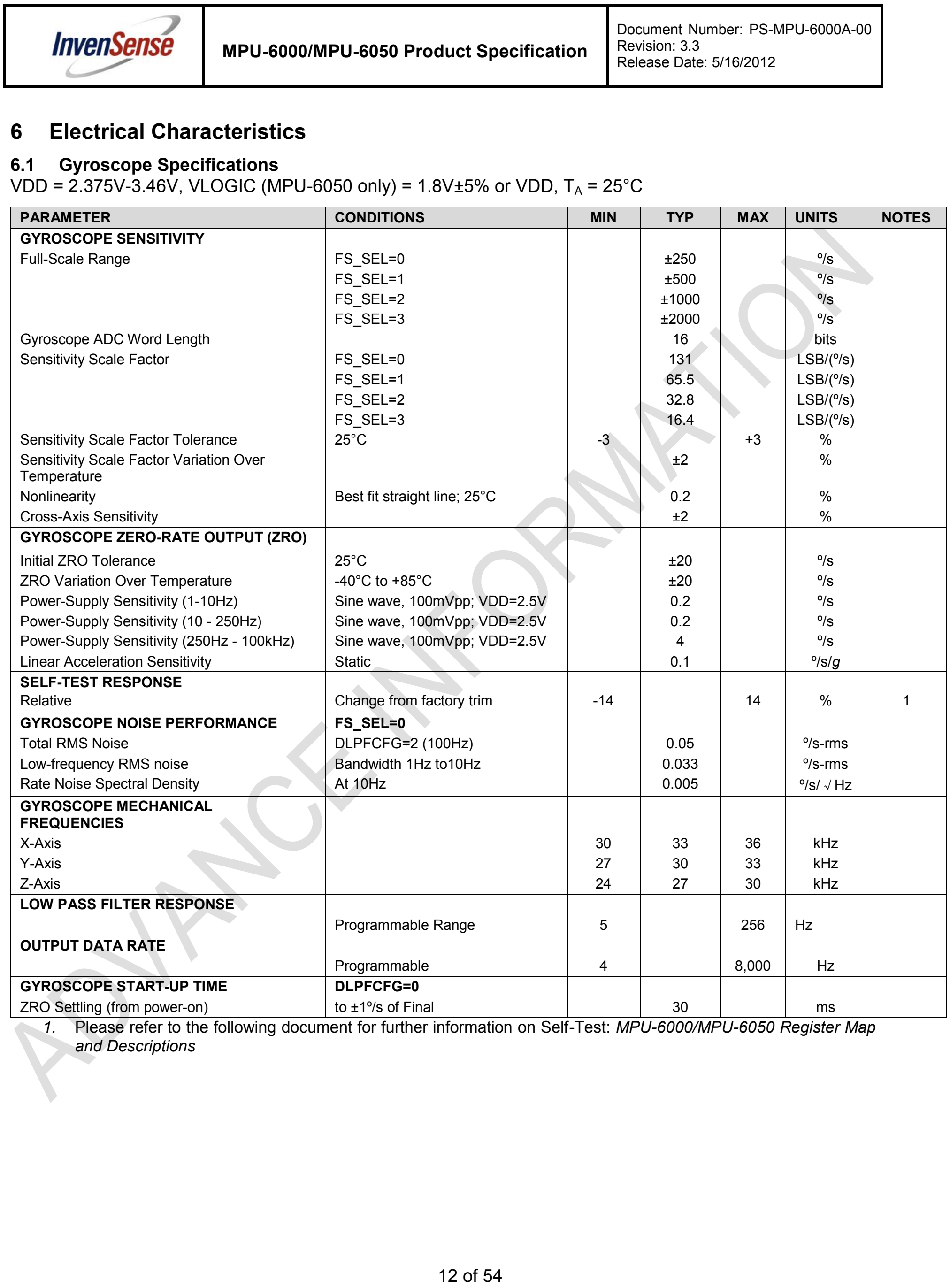
Communication with all registers of the device is performed using either I^{2}C at 400kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with ±1% variation over the operating temperature range.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the MPU-60X0 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000*g* shock tolerance, and has programmable low-pass filters for the gyroscopes, accelerometers, and the on-chip temperature sensor.

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I^{2}C interface. The VLOGIC voltage may be 1.8V±5% or VDD.

The MPU-6000 and MPU-6050 are identical, except that the MPU-6050 supports the I^{2}C serial interface only, and has a separate VLOGIC reference pin. The MPU-6000 supports both I^{2}C and SPI interfaces and has a single supply pin, VDD, which is both the device’s logic reference supply and the analog supply for the part. The table below outlines these differences:

7 of 54



InvenSense

MPU-6000/MPU-6050 Product Specification

Document Number: PS-MPU-6000A-00
Revision: 3.3
Release Date: 5/16/2012

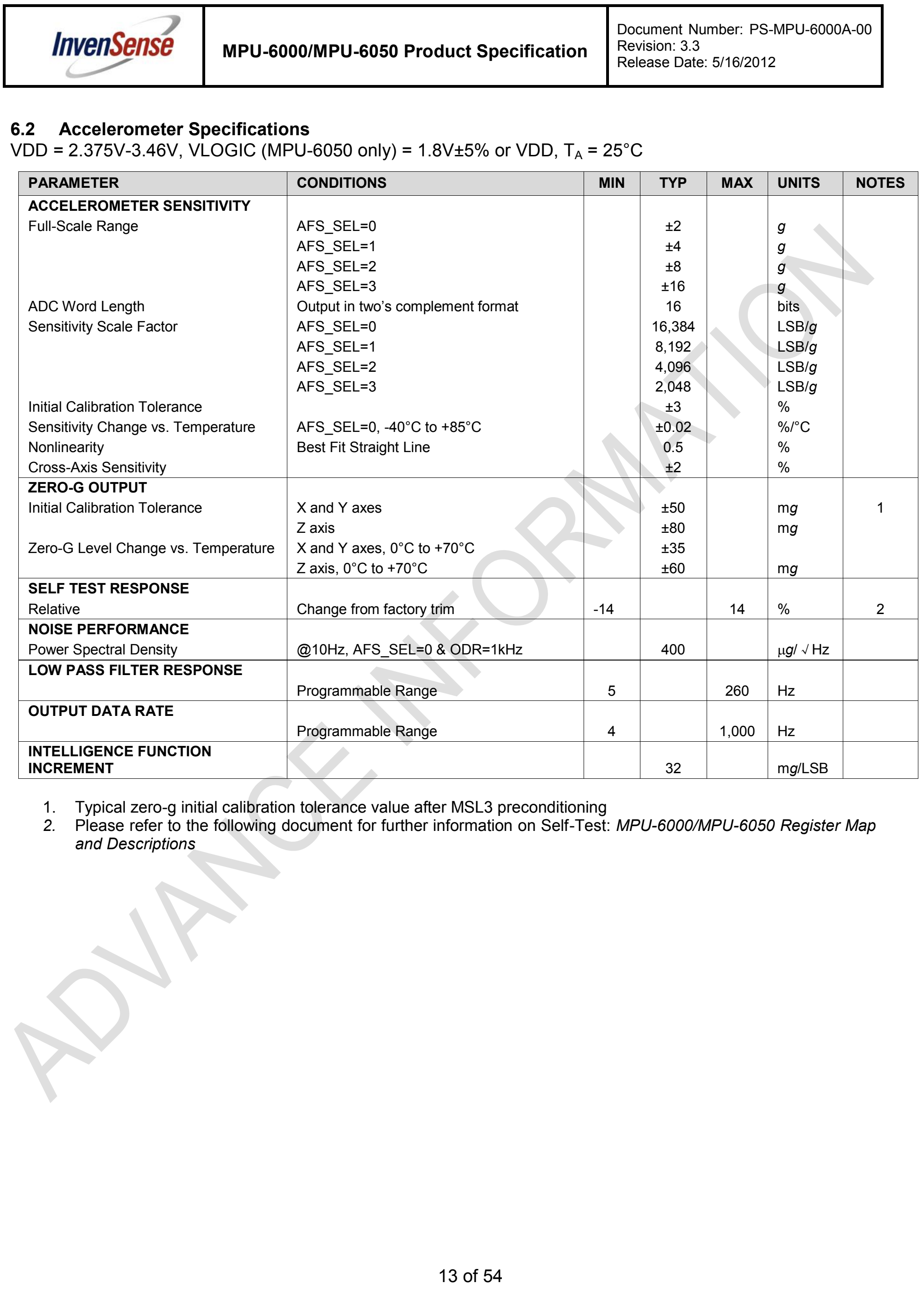
6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/*g*	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	**FS_SEL=0**					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/ √ Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		256	Hz	
OUTPUT DATA RATE						
	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME	**DLPFCFG=0**					
ZRO Settling (from power-on)	to ±1°/s of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



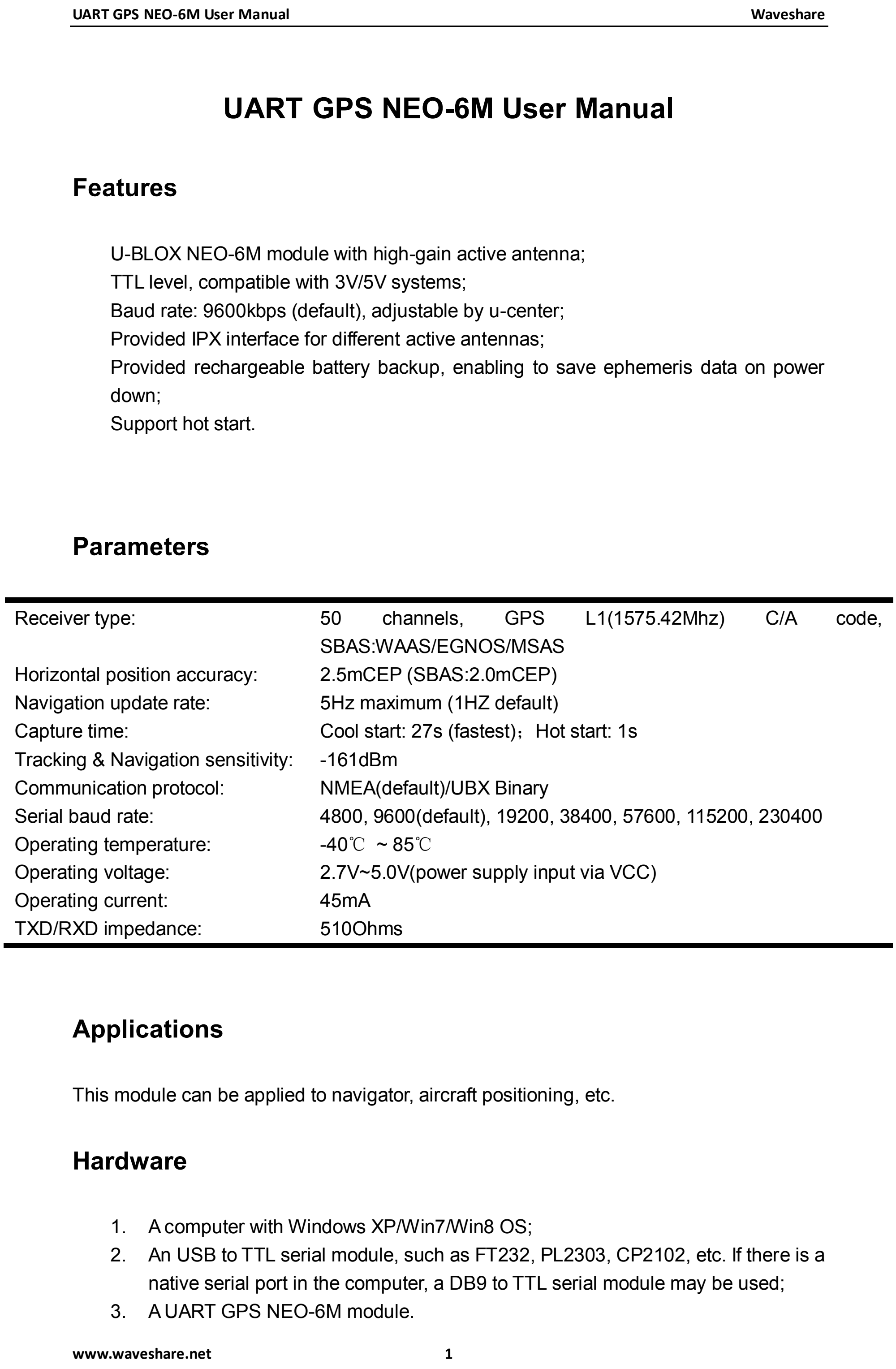
InvenSense | MPU-6000/MPU-6050 Product Specification | Document Number: PS-MPU-6000A-00
Revision: 3.3
Release Date: 5/16/2012

6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		*g*	
	AFS_SEL=1		±4		*g*	
	AFS_SEL=2		±8		*g*	
	AFS_SEL=3		±16		*g*	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/*g*	
	AFS_SEL=1		8,192		LSB/*g*	
	AFS_SEL=2		4,096		LSB/*g*	
	AFS_SEL=3		2,048		LSB/*g*	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		m*g*	1
	Z axis		±80		m*g*	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		m*g*	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		μ*g*/ √Hz	
LOW PASS FILTER RESPONSE	Programmable Range	5		260	Hz	
OUTPUT DATA RATE	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		m*g*/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



UART GPS NEO-6M User Manual

Features

U-BLOX NEO-6M module with high-gain active antenna;
TTL level, compatible with 3V/5V systems;
Baud rate: 9600kbps (default), adjustable by u-center;
Provided IPX interface for different active antennas;
Provided rechargeable battery backup, enabling to save ephemeris data on power down;
Support hot start.

Parameters

Receiver type:	50 channels, GPS L1(1575.42Mhz) C/A code, SBAS:WAAS/EGNOS/MSAS
Horizontal position accuracy:	2.5mCEP (SBAS:2.0mCEP)
Navigation update rate:	5Hz maximum (1HZ default)
Capture time:	Cool start: 27s (fastest)；Hot start: 1s
Tracking & Navigation sensitivity:	-161dBm
Communication protocol:	NMEA(default)/UBX Binary
Serial baud rate:	4800, 9600(default), 19200, 38400, 57600, 115200, 230400
Operating temperature:	-40℃ ~ 85℃
Operating voltage:	2.7V~5.0V(power supply input via VCC)
Operating current:	45mA
TXD/RXD impedance:	510Ohms

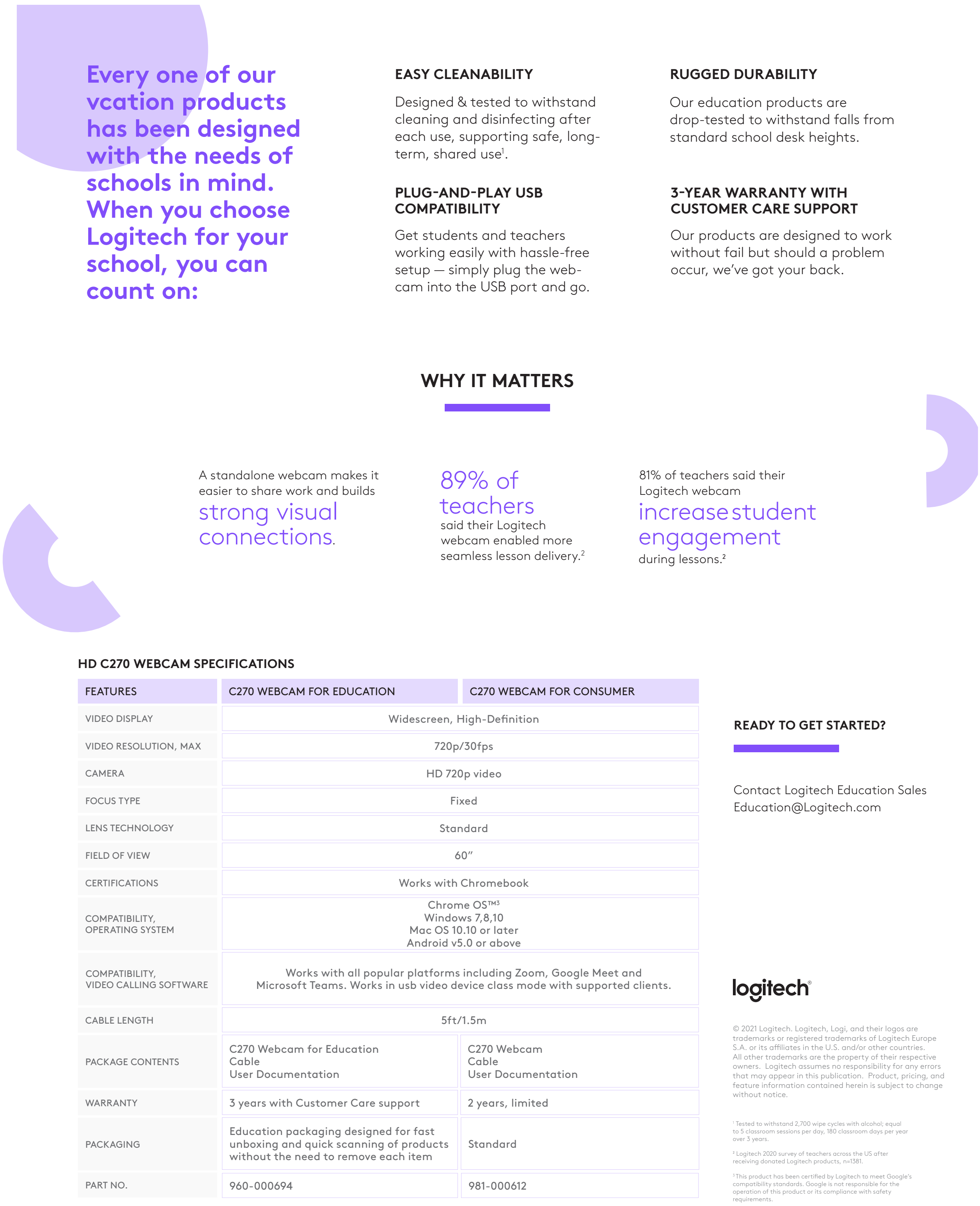
Applications

This module can be applied to navigator, aircraft positioning, etc.

Hardware

1. A computer with Windows XP/Win7/Win8 OS;
2. An USB to TTL serial module, such as FT232, PL2303, CP2102, etc. If there is a native serial port in the computer, a DB9 to TTL serial module may be used;
3. A UART GPS NEO-6M module.

APPENDIX



Every one of our vcation products has been designed with the needs of schools in mind. When you choose Logitech for your school, you can count on:

EASY CLEANABILITY

Designed & tested to withstand cleaning and disinfecting after each use, supporting safe, long-term, shared use[1].

PLUG-AND-PLAY USB COMPATIBILITY

Get students and teachers working easily with hassle-free setup — simply plug the web-cam into the USB port and go.

RUGGED DURABILITY

Our education products are drop-tested to withstand falls from standard school desk heights.

3-YEAR WARRANTY WITH CUSTOMER CARE SUPPORT

Our products are designed to work without fail but should a problem occur, we've got your back.

WHY IT MATTERS

A standalone webcam makes it easier to share work and builds **strong visual connections.**

89% of teachers said their Logitech webcam enabled more seamless lesson delivery.[2]

81% of teachers said their Logitech webcam **increase student engagement** during lessons.[2]

HD C270 WEBCAM SPECIFICATIONS

FEATURES	C270 WEBCAM FOR EDUCATION	C270 WEBCAM FOR CONSUMER
VIDEO DISPLAY	Widescreen, High-Definition	
VIDEO RESOLUTION, MAX	720p/30fps	
CAMERA	HD 720p video	
FOCUS TYPE	Fixed	
LENS TECHNOLOGY	Standard	
FIELD OF VIEW	60”	
CERTIFICATIONS	Works with Chromebook	
COMPATIBILITY, OPERATING SYSTEM	Chrome OS™[3] Windows 7,8,10 Mac OS 10.10 or later Android v5.0 or above	
COMPATIBILITY, VIDEO CALLING SOFTWARE	Works with all popular platforms including Zoom, Google Meet and Microsoft Teams. Works in usb video device class mode with supported clients.	
CABLE LENGTH	5ft/1.5m	
PACKAGE CONTENTS	C270 Webcam for Education Cable User Documentation	C270 Webcam Cable User Documentation
WARRANTY	3 years with Customer Care support	2 years, limited
PACKAGING	Education packaging designed for fast unboxing and quick scanning of products without the need to remove each item	Standard
PART NO.	960-000694	981-000612

READY TO GET STARTED?

Contact Logitech Education Sales
Education@Logitech.com

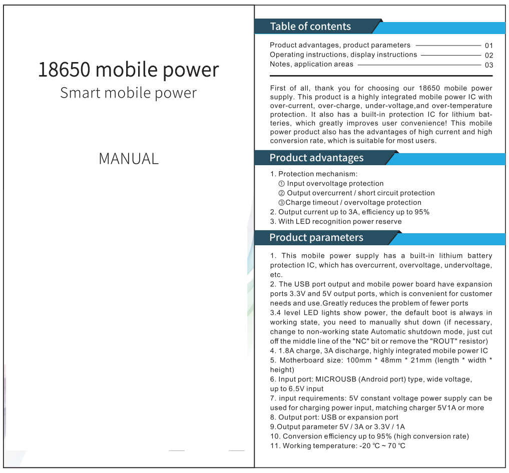
logitech®

[1] Tested to withstand 2,700 wipe cycles with alcohol; equal to 5 classroom sessions per day, 180 classroom days per year over 3 years.

[2] Logitech 2020 survey of teachers across the US after receiving donated Logitech products, n=1381.

[3] This product has been certified by Logitech to meet Google's compatibility standards. Google is not responsible for the operation of this product or its compliance with safety requirements.

APPENDIX



18650 mobile power

Smart mobile power

MANUAL

Table of contents

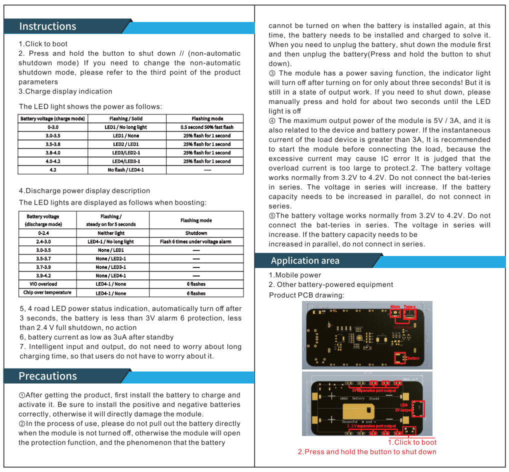
First of all, thank you for choosing our 18650 mobile power supply. This product is a highly integrated mobile power IC with over-current, over-charge, under-voltage,and over-temperature protection. It also has a built-in protection IC for lithium batteries, which greatly improves user convenience! This mobile power product also has the advantages of high current and high conversion rate, which is suitable for most users.

Product advantages

1. Protection mechanism:
   ① Input overvoltage protection
   ② Output overcurrent / short circuit protection
   ③Charge timeout / overvoltage protection
2. Output current up to 3A, efficiency up to 95%
3. With LED recognition power reserve

Product parameters

1. This mobile power supply has a built-in lithium battery protection IC, which has overcurrent, overvoltage, undervoltage, etc.
2. The USB port output and mobile power board have expansion ports 3.3V and 5V output ports, which is convenient for customer needs and use.Greatly reduces the problem of fewer ports
3.4 level LED lights show power, the default boot is always in working state, you need to manually shut down (if necessary, change to non-working state Automatic shutdown mode, just cut off the middle line of the "NC" bit or remove the "ROUT" resistor)
4. 1.8A charge, 3A discharge, highly integrated mobile power IC
5. Motherboard size: 100mm * 48mm * 21mm (length * width * height)
6. Input port: MICROUSB (Android port) type, wide voltage,
up to 6.5V input
7. input requirements: 5V constant voltage power supply can be used for charging power input, matching charger 5V1A or more
8. Output port: USB or expansion port
9.Output parameter 5V / 3A or 3.3V / 1A
10. Conversion efficiency up to 95% (high conversion rate)
11. Working temperature: -20 ℃ ~ 70 ℃



Instructions

1.Click to boot
2. Press and hold the button to shut down // (non-automatic shutdown mode) If you need to change the non-automatic shutdown mode, please refer to the third point of the product parameters
3.Charge display indication

The LED light shows the power as follows:

Battery voltage (charge mode)	Flashing / Solid	Flashing mode
0-3.0	LED1 / No long light	0.5 second 50% fast flash
3.0-3.5	LED1 / None	25% flash for 1 second
3.5-3.8	LED2 / LED1	25% flash for 1 second
3.8-4.0	LED3/LED2-1	25% flash for 1 second
4.0-4.2	LED4/LED3-1	25% flash for 1 second
4.2	No flash / LED4-1	—

4.Discharge power display description

The LED lights are displayed as follows when boosting:

Battery voltage (discharge mode)	Flashing / steady on for 5 seconds	Flashing mode
0-2.4	Neither light	Shutdown
2.4-3.0	LED4-1 / No long light	Flash 6 times under voltage alarm
3.0-3.5	None / LED1	—
3.5-3.7	None / LED2-1	—
3.7-3.9	None / LED3-1	—
3.9-4.2	None / LED4-1	—
VIO overload	LED4-1 / None	6 flashes
Chip over temperature	LED4-1 / None	6 flashes

5, 4 road LED power status indication, automatically turn off after 3 seconds, the battery is less than 3V alarm 6 protection, less than 2.4 V full shutdown, no action
6, battery current as low as 3uA after standby
7. Intelligent input and output, do not need to worry about long charging time, so that users do not have to worry about it.

Precautions

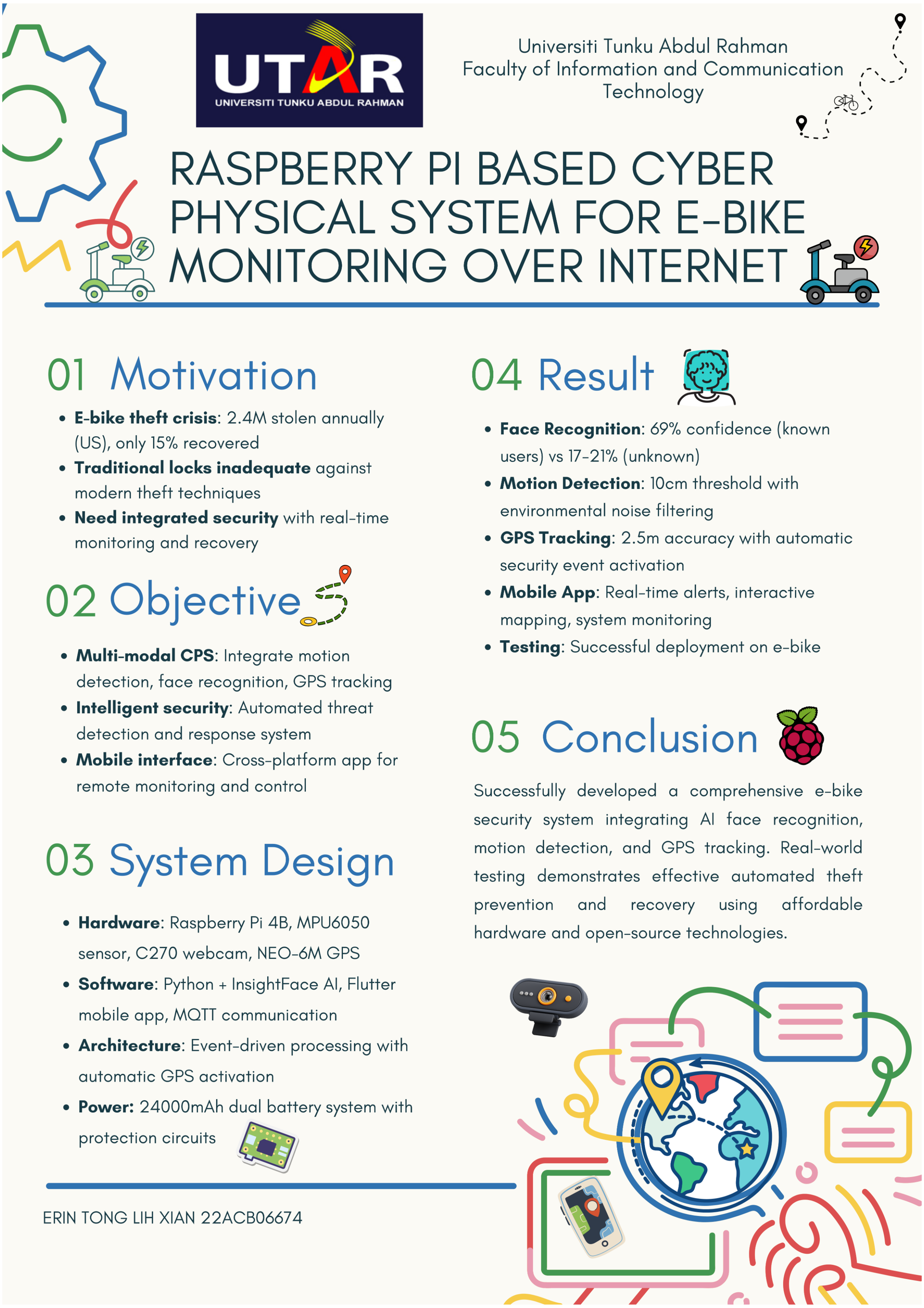
①After getting the product, first install the battery to charge and activate it. Be sure to install the positive and negative batteries correctly, otherwise it will directly damage the module.
②In the process of use, please do not pull out the battery directly when the module is not turned off, otherwise the module will open the protection function, and the phenomenon that the battery cannot be turned on when the battery is installed again, at this time, the battery needs to be installed and charged to solve it. When you need to unplug the battery, shut down the module first and then unplug the battery(Press and hold the button to shut down).
③ The module has a power saving function, the indicator light will turn off after turning on for only about three seconds! But it is still in a state of output work. If you need to shut down, please manually press and hold for about two seconds until the LED light is off
④ The maximum output power of the module is 5V / 3A, and it is also related to the device and battery power. If the instantaneous current of the load device is greater than 3A, It is recommended to start the module before connecting the load, because the excessive current may cause IC error It is judged that the overload current is too large to protect.2. The battery voltage works normally from 3.2V to 4.2V. Do not connect the bat-teries in series. The voltage in series will increase. If the battery capacity needs to be increased in parallel, do not connect in series.
⑤The battery voltage works normally from 3.2V to 4.2V. Do not connect the bat-teries in series. The voltage in series will increase. If the battery capacity needs to be
increased in parallel, do not connect in series.

Application area

1.Mobile power
2. Other battery-powered equipment
Product PCB drawing:

1.Click to boot
2.Press and hold the button to shut down

POSTER



Universiti Tunku Abdul Rahman
Faculty of Information and Communication Technology

RASPBERRY PI BASED CYBER PHYSICAL SYSTEM FOR E-BIKE MONITORING OVER INTERNET

01 Motivation

- **E-bike theft crisis**: 2.4M stolen annually (US), only 15% recovered
- **Traditional locks inadequate** against modern theft techniques
- **Need integrated security** with real-time monitoring and recovery

02 Objective

- **Multi-modal CPS**: Integrate motion detection, face recognition, GPS tracking
- **Intelligent security**: Automated threat detection and response system
- **Mobile interface**: Cross-platform app for remote monitoring and control

03 System Design

- **Hardware**: Raspberry Pi 4B, MPU6050 sensor, C270 webcam, NEO-6M GPS
- **Software**: Python + InsightFace AI, Flutter mobile app, MQTT communication
- **Architecture**: Event-driven processing with automatic GPS activation
- **Power:** 24000mAh dual battery system with protection circuits

ERIN TONG LIH XIAN 22ACB06674

04 Result

- **Face Recognition**: 69% confidence (known users) vs 17-21% (unknown)
- **Motion Detection**: 10cm threshold with environmental noise filtering
- **GPS Tracking**: 2.5m accuracy with automatic security event activation
- **Mobile App**: Real-time alerts, interactive mapping, system monitoring
- **Testing**: Successful deployment on e-bike

05 Conclusion

Successfully developed a comprehensive e-bike security system integrating AI face recognition, motion detection, and GPS tracking. Real-world testing demonstrates effective automated theft prevention and recovery using affordable hardware and open-source technologies.