

SEMI-AUTONOMOUS 4 WHEEL ROBOT

By

Chia Zheng Lin

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirements

For the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

APRIL 2014

DECLARATION OF ORIGINALITY

I declare that this report entitled “**SEMI-AUTONOMOUS 4 WHEEL ROBOT**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Mr. W.K. Lee for giving me this bright opportunity to engage in a robotic project. It is my first step to establish a career in robotic field. A million thanks to you.

To all my friends and foes, for their support and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

This project is a robotic project which develops a complete system for a semi-autonomous 4 wheel robot. The robot will be able to traverse in a structured area with the ability to avoid obstacle and perform some simple instructions.

Few types of sensors, components and devices are implemented onto the robot along with a wireless connection system for the robot to operate semi-autonomously. The limitations of this project is its semi-autonomous, flat operating environment and being a prototype platform.

The robot aims to achieve the following 3 objective: To gain information about its operating environment, to move itself throughout its operating environment with very less human assistance and to work for an extended period without human intervention.

The Microchip dsPIC technology is chosen as the Center Processing Unit of the robot, along with a DC motor controller, ultrasonic sensor, Inertial Measurement Unit, and RF communication module.

The final output of the project would be the robot prototype with a basic complete semi-autonomous system.

TABLE OF CONTENTS

TITLE	i
DECLARATION OF ORIGINALITY	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES	vii
LIST OF TABLES.....	ix
LIST OF ABBREVIATIONS.....	x
CHAPTER 1. INTRODUCTION	1
1.1 Motivation and Problem Statement.....	1
1.2 Project Background.....	3
1.3 Objectives	5
CHAPTER 2. LITERATURE REVIEW AND FINDINGS	7
2.1 A Human-Like Semi-Autonomous Mobile Security Robot.....	7
2.2 Mobile Robots for Search and Rescue	9
CHAPTER 3. TOOLS AND COMPONENTS.....	12
3.1 Product Design.....	12
3.2 Software and Hardware Tools	15
3.2.1 Integrated Development Environment (IDE).....	15
3.2.2 Compiler	16
3.2.3 Programmer.....	17
3.2.4 Oscilloscope.....	19
3.3 Product Components.....	21
3.3.1 Center Processing Unit	21
3.3.2 Motor controller	24
3.3.3 Position detection sensor.....	25

3.3.4 Range detection sensor.....	27
3.3.5 Radio Frequency (RF) communication.....	28
CHAPTER 4. METHODOLOGY AND APPLICATION	30
4.1 Analog-to-Digital Converter.....	32
4.2.1 Ultrasonic Range Finder	36
4.2.2 Inertial Measurement Unit	37
4.3 Pulse Wave Modulation.....	38
4.3.1 Servomotors	40
4.4 RF Transceiver.....	41
4.4.1 RF Implementation	46
4.5 Information Output	48
CHAPTER 5. PROJECT REVIEW	49
5.1 Project Discussion and Conclusion.....	49
5.2 Future Work.....	50
BIBLIOGRAPHY	51
APPENDICES A. REGISTER MAP OF NRF24L01+ FROM DATA SHEET	A

LIST OF FIGURES

Figure Number	Title	Page
1.1	A picture of the Mars Rover	4
1.2	A picture of the Stealth UAV	4
2.1	Side view and front view of MARVIN	7
2.2	The new MARVIN	8
2.3	Mapping robot	9
2.4	Human following robot	9
2.5	The PeLoTe experiment in the firefighter training house	11
3.1	Top view of the robot prototype	13
3.2	Side view of the robot prototype	13
3.3	Pin connection diagram	14
3.4	Overview of MPLAB X IDE	15
3.5	Overview of UIC00B Programmer	17
3.6	Overview of PICkit 2 software UI	18
3.7	Overview of PicoScope 6 software UI	20
3.8	Overview of PicoScope handheld oscilloscope	20
3.9	Overview of SKsd40A attached with dsPIC30F4013 and a LCD	23
3.10	Pin diagram of the dsPIC30F4013 microcontroller	23
3.11	Overview of L298N Dual H Bridge Motor Driver controller module	24
3.12	Overview of SN-IMU5D-LC	25
3.13	Overview of LV-MaxSonar-EZ1	27

3.14	Overview of nFR24L01+ module	28
4.1	Operation flow of the robot	31
4.2	Diagram of nFR24L01+	41

LIST OF TABLES

Table Number	Title	Page
4.1	Port Register Map	35
4.2	ADC Register Map	35
4.3	Output Compare Register Map	39
4.4	Serial Peripheral Interface Register Map	42

LIST OF ABBREVIATIONS

USB	Universal Serial Bus
RF	Radio Frequency
Tx	Transmit
Rx	Receive
LCD	Liquid-Crystal Display
LED	Light Emitting Device
SW	Switch
L1	Left side Forward
L0	Left side Reverse
R1	Right side Forward
R0	Right side Reverse
PWM	Pulse Wave Modulation
IDE	Integrated Development Environment
DOF	Depth of Field
IMU	Inertial Measurement Unit
ADC	Analog-to-Digital Converter
ADPCFG	AD Port Configuration register
ADCHS	AD Channel Select register
ADCSSL	AD Input Scan Selection register
ADCON	AD Control register
ADCBUF	AD Buffer
PR	Period Register
OC	Output Compare
OCxRS	Output Compare Second Register

OCxR	Output Compare Main Register
OCxCON	Output Compare Configuration register
IRQ	Interrupt Request
MISO	Master In Slave Out
MOSI	Master Out Slave In
CE	Chip Enable
CSN	Chip Select NOT
SPI	Serial Peripheral Interface
SPIxSTAT	SPI Status register
SPIROV	SPI Overflow register
SPIEN	SPI Enable register
MSTEN	Master mode Enable register
GFSK	Gaussian Frequency Shift Keying
CRC	Cyclic Redundancy Check
ACK	Acknowledgement
ART	Auto Re-Transmit
LNA	Low Noise Amplifier
TxCON	Timer Configuration register

CHAPTER 1. INTRODUCTION

The present project is aimed to build a semi-autonomous robot which is able to navigate itself and perform some simple tasks semi-autonomously in a building.

1.1 Motivation and Problem Statement

The ability of an autonomous robot to move itself throughout its operating environment is usually unpredictable and unstructured. This is considered one of the biggest challenge in this project. The robot has to scan, check for its surrounding and determine the next move. Ideally, the solution to the problem should be both deliverable and affordable.

To increase the effectiveness and reliability of the robot, the size is limited to be not too big which will consume spaces and will be more flexible in moving itself to perform tasks. Some weaknesses and limitations of the project's product are listed below:

- Semi-autonomous:
The robot will be designed for semi-autonomous control. It will not be fully autonomous and operate by itself because of its limited functions. During its performances, certain difficult processes will still be needing the static instructions from human. This is why the RF communication will be implemented.

- Flat operating environment surface:

With its broad and low chassis, this 4 wheel robot is best designed to be operating on a flat and smooth surface. Although with the ability to avoid obstacles, it might not be able to move nicely on an uneven surface such as stairway, hill, and etc.

- Prototype platform:

For being a prototype design semi-autonomous robot, its system will only consist some of the most basic functions of an autonomous robot with a simple controlling method. Further upgrade and improvising will only be done after completing its primary objectives.

1.2 Project Background

Robots are physical agents and mechanical devices guided by electronic circuits or computer programs to perform one to many autonomous or preprogrammed tasks in the physical world. Nowadays with the fast paced of technological advancement and development, human become lazier and dependent on technology. The robots which are able to perform different kind of tasks are highly in demand. They have replace human in many places, especially for tasks that will put human life in risk.

Robots can be controlled manually, fully autonomous, and semi-autonomously. While other robots require instructions received from human from time to time in performing scheduled tasks, autonomous robots can perform tasks (mostly in unstructured environments) without continuous guide and instructions sent by human. Hence it can decrease the needs of man power and save many time.

There are also many types of robots which have different functions or usage in different area. For example, mobile robot, industrial robot, educational robot, modular robot, and collaborative robots. Having different functions, each type of robot is important to human in different aspect. For example, industrial robot can provide much systematic and faster production speed without many mistakes than human workers.

For discovery or protection, robots can too be very useful. There are underwater robot which can advances the human technology in the sea exploration or deep sea development and mining. Flying robots provide human a better experience and knowledge towards our sky, or even the space.

With the fast advancement of technology after the 3rd industry revolution and the finishing of natural resource and supply from earth, human will move into the space. Robots will provide greater opportunities for human to explore the unknown space or the planets on the far distance. For example, the Mars Rover aided human in exploring the Mars.

There is so many achievements which human could now accomplish alone without the help of their technology. Therefore, as one of the masterpiece in human technology advancement, robot will be one of the best way to assist human in obtaining more knowledge of the future and the unknown world.

The targeted users of this project's final delivery range from home users to industry users. Robot has always show their usefulness in many places. Therefore, in the near future, it will become a common part of everyone's life.

The targeted main contribution of this project is to have an affordable complete system for the semi-autonomous robot to be common in the world. The robots will be able to replace many traditional methods and bring the robotic technologies a big step forward in the future.



Figure 1.1. A picture of the Mars Rover



Figure 1.2. A picture of the Stealth UAV

1.3 Objectives

The overarching aim of this project is to develop a complete system for a semi-autonomous 4 wheel robot. The robot will be able to traverse in office building with the ability to avoid obstacles and perform some simple tasks.

It can operate in fully or semi-autonomous mode. Specifically, it aims to achieve the following objectives:

- To gain information about its operating environment:

The robot will be able to survey its surrounding and detect obstacles accurately with the use of its sensors. For example, when the robot is approaching a wall, its sensors will be able to detect the presence of the wall before getting in touch with it. Or when the robot is going downhill on a sloppy surface, its sensors will be able to provide information about the position of the robot

- To move itself throughout its operating environment with very less human assistance:

The robot will be able to utilize the obtained information about its operating environment in order to perform countermeasures with its effectors. In other words, the presence of wall detected earlier will enforce the robot to modify its moving direction to avoid getting in touch with the wall by making a turn. Or the robot will slow down its speed when going downhill. However, human assistance is need for paths that are difficult for the robot to navigate, example staircase, ramp and etc.

- To work for an extended period without human intervention:

With its semi-autonomous mode, the robot will be able to continue its operation and perform the given tasks without having to be manipulated continuously by human, For example, after receiving some simple instructions form human, the robot will be able to complete the tasks without the needs of further intervention from human.

CHAPTER 2. LITERATURE REVIEW AND FINDINGS

Variety of autonomous robot have been developed since the past for different purposes. Most autonomous robots are implemented for rescue mission, medical assistance, map exploration, factory works, environment cleaning, security and etc. Considering being a prototype project, only some related works which had done in the past will be discussed in this section.

2.1 A Human-Like Semi-Autonomous Mobile Security Robot

Department of Physics and Electronic Engineering in University of Waikato designed MARVIN (Mobile Autonomous Robotic Vehicle for Indoor Navigation) in December, 2004. The robot is created to be an indoor security agent that can interact with human with its human-like face.



Figure 2.1. Side view and front view of MARVIN

MARVIN is implemented with various sensors to avoid obstacles. For long range sensing, a custom-developed laser range finder which operates by projecting laser through a cylinder lens was built-in. The reflected beam is then processed and converted into a range value.

Intermediate ranging takes the form of a position sensitive devices instead of the initial infrared intensity based sensor due to the result's dependence on the color, orientation and texture of the reflecting surface.

For short range detection, contact sensors were placed at the bottom corner of MARVIN. These sensors trigger a contact switch if activated when they contacted with some surface although the long and intermediate range sensors would have already detected the obstacles and initiated some avoidance processes.



Figure 2.2. The new MARVIN

2.2 Mobile Robots for Search and Rescue

According to a study published on the International Workshop on Safety, Security and Rescue Robotics, the design and implementation of a semi-autonomous mobile robot are further discussed focusing on the robot entities in the PeLoTe project (Building Presence through Localization for Hybrid Telematics Teams).

In the project, the mobile robots have two main functions during a mission; one is to follow the human entity and another one is to provide the sensor data for mapping system. The robot tasks is to provide data for map building, explore the environment, search for the wounded in wide areas and carry equipment for other human rescuers.

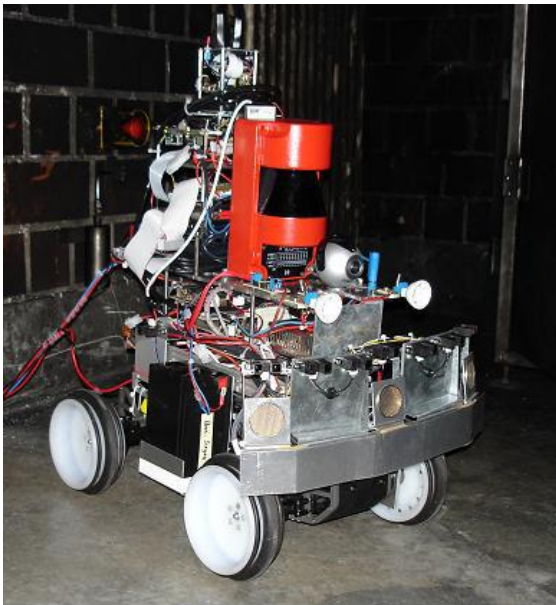


Figure 2.3. Mapping robot

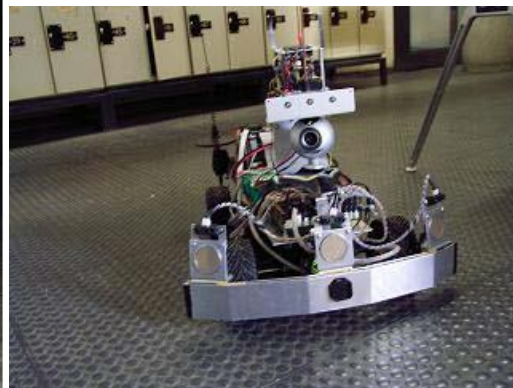


Figure 2.4. Human following robot

The robot (Mobile Experimental Robot for Locomotion and Intelligent Navigation, MERLIN) is based on the existing robot basic structure and technologies which is controlled by 80C167 CR 16 bit processor and PC104.

Two motors are the servo steering motor for changing directions and DC driving motor for the robot orientation and speed control. Implemented components are encoders, infrared sensors, ultrasonic sensors, 3-Axis magnetic compass, laser scanner, gyroscope, beacon system, bumpers, web camera, wireless communication system, and white LED lamps.

MERLIN is remotely controlled using TCP/IP communication with programmed client and server using java software. There are different layers of module implemented in the software structure of the robot:

- Server interface module: This module is responsible for establishing connection to the PeLoTe server and keep the communication connected. In case of connection lost, it will inform all modules and try to reestablish the connection.
- Decision module: The function of this module is to interpret the path data from path planner and notification of robot events and make initial path decision.
- Mapping module: To support the map updating and exploration, the mapping module builds map from data obtained from its sensors and the maps are sent to the server for efficient planning.
- Planning module: This module takes the plan from decision module and mapping module, then generates better plans based on the obstacles detected and send them to movement control module.
- Localization module: This module is responsible for the localization of the robot, and also handles the corrections of the robot position from the operator.

- Direction movement control module: In some cases when the operator may need to manually control the robot using external controller, this module handles the communication between the robot and the operator.
- Diagnostic module: This module informs the server and operators about the robot status. For example, battery level, and sensors data such as temperature, detected gas.
- Movement control module: This module translate the trajectory plan from the planning module into robot movement commands and send to the robot.
- Bumper module: This module detects obstacles and informs the movement control module to stop the current movement command.

An experiment were performed in the firefighter training house using the PeLoTe equipment. It was demonstrated and concludes that semi-autonomous robots may improve the quality of the whole rescue mission, mainly in speed, coverage area exploration and reduce danger to the rescue team.



Figure 2.5. The PeLoTe experiment in the firefighter training house of Wurzburg, Germany

CHAPTER 3. TOOLS AND COMPONENTS

3.1 Product Design

The basic structure for the robot is a metallic chassis with a dimension of 25.5 x 26.8 cm and 4 rubber wheels attached to 4 servomotors.

On top of the flat surface of the chassis, a SKds40A kit is implemented along with a Microchip dsPIC30F4013 microcontroller is attached as the center processing unit of the robot. The L298N motor driver is implemented to control the 4 servomotor wheels.

A Maxsonar EZ-1 ultrasonic sensor which serves as the main navigational sensor of the robot is attached on the preset front side of the robot facing outside. A SN-IMU5D-LC Inertial Measurement Unit which incorporates 2 Gyro Sensors and a 3-Axis Accelerometer is attached on the middle part of the robot for robot movement and direction monitoring.

A RF-TRX-2.4G 2.4GHz wireless RF transceiver is implemented as the main communication protocol between the robot and human. The power source of the robot will be the attachable 7.4V Li-poly RC battery on the rear side.

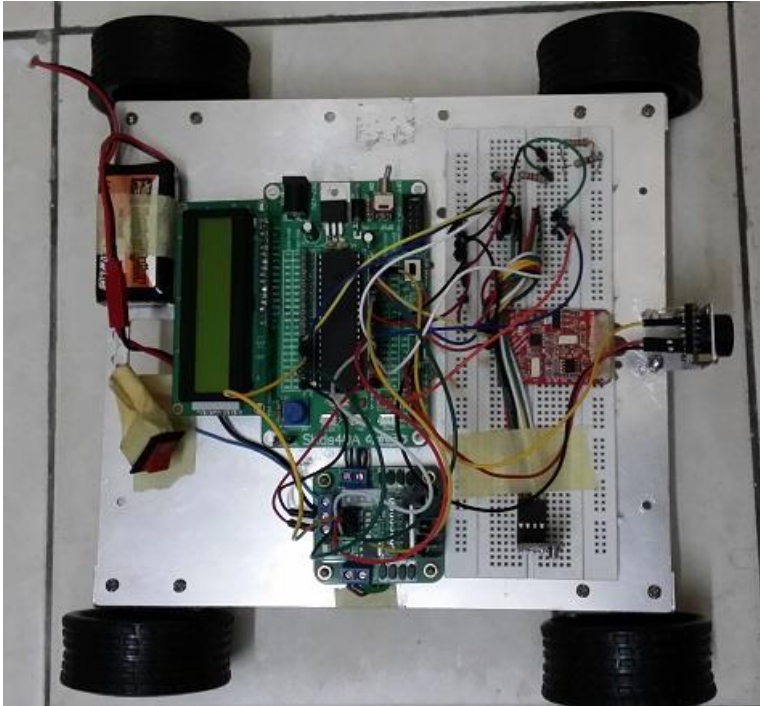


Figure 3.1. Top view of the robot prototype

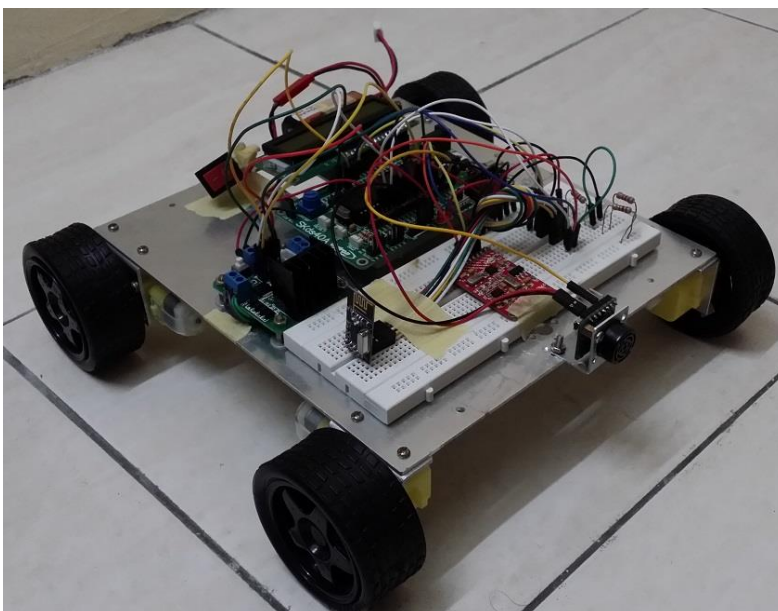


Figure 3.2. Side view of the robot prototype

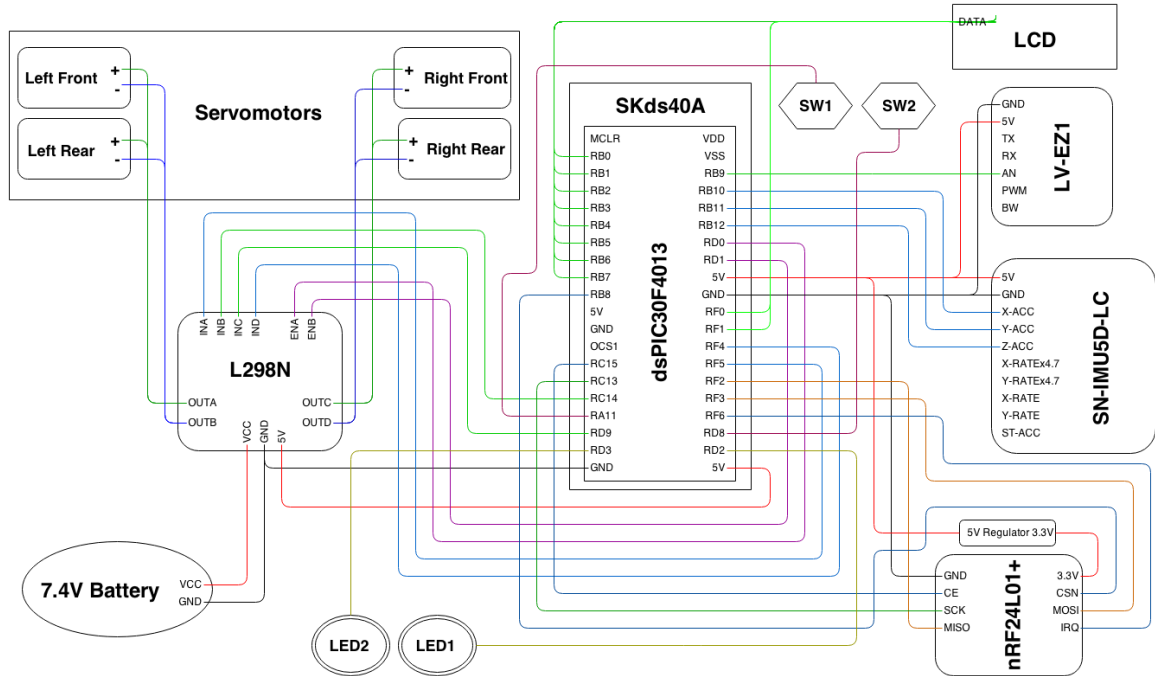


Figure 3.3. Pin connection diagram

3.2 Software and Hardware Tools

3.2.1 Integrated Development Environment (IDE)

Due to the implementation of Microchip dsPIC30F4013 microcontroller, the MPLAB X provided by Microchip is chosen as the main IDE to develop applications for the embedded microcontroller.

MPLAB X is specially designed for the Microchip family's product. It is based on the Oracle open source Net Beans IDE. Some of the features supported by the MPLAB X are useful for developing this project:

- Supports multiple configurations within projects
- Supports multiple versions of the same compiler
- Supports multiple Debug tools for the same type
- Supports hyperlinks for fast navigation to declarations and includes

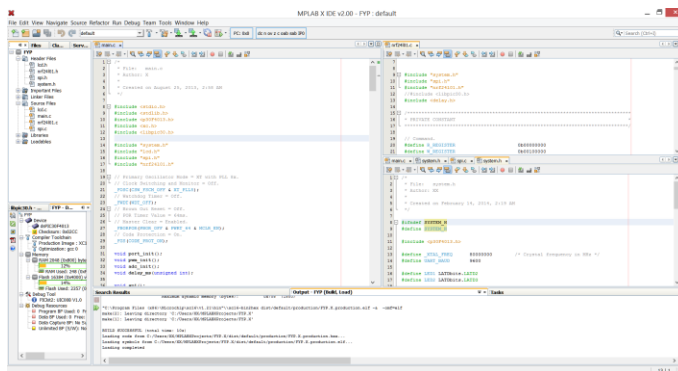


Figure 3.4. Overview of MPLAB X IDE

3.2.2 Compiler

Out of many compilers, the MPLAB XC16 C compiler is chosen as the main compiler for MPLAB X. The XC16 C compiler supports all the 16-bit devices from Microchip, hence it is compatible for 16-bit dsPIC30F4013 microcontroller which is used in this project.

The MPLAB XC16 C compiler translates standard ANSI C programs into 16-bit device assembly language source, and it also supports many language extensions and command-line options which allow full access to the 16-bit device hardware capabilities. It also affords good control for the compiler code generator.

The MPLAB XC16 C compiler ports from the GNU Compiler Collection (GCC) compiler by the Free Software Foundation and is also available for several popular operating systems, including Linux, Apple OS X, and 32 and 64-bit Windows.

The compiler offers three operating modes: Free, Standard or PRO. This project uses the free mode which is available for unlicensed customers.

3.2.3 Programmer

The hardware device used as the programmer to communicate with the SKds40A kit is the UIC00B programmer obtained from Cytron Technologies. It is designed to program popular Flash PIC MCU including most of the Microchip PIC family in 8, 16 and 32-bit. It is a truly portable programmer which offers many capabilities and features:

- USB Plug and Play function
- ICSP connection
- On-board programming
- External power source is not required to function
- Optional socket to program 18, 28, and 40 pins PIC microcontroller
- Provides output for +3.3V or +5V PIC

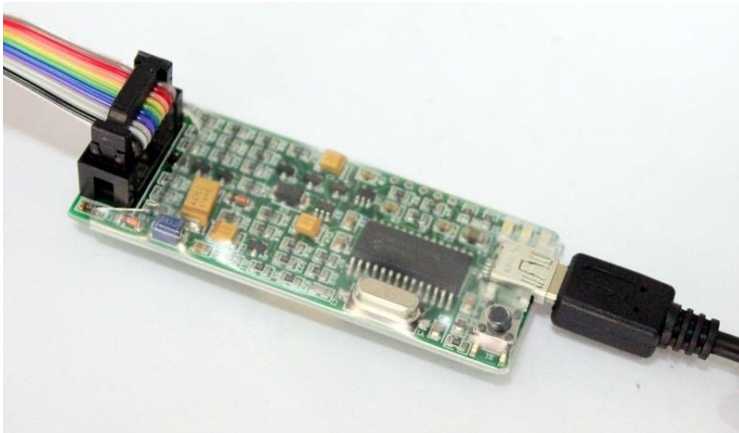


Figure 3.5. Overview of UIC00B Programmer (connected with USB and rainbow cable)

The UIC00B programmer is compatible with the Microchip PICkit 2 Programmer application which is also compatible with the MPLAB X IDE. Therefore, this project uses the Microchip PICkit 2 Programmer application to work with the UIC00B programmer.

The PICkit 2 Programmer application allows many devices which includes the hardware used in this project to be programmed. User can manually import or export hex file into or from the connected devices, or uses the MPLAB X IDE to automatically upload the compiled hex file into the SKds40A kit through the UIC00B. The SKds40A kit can be provided with 5V power by the UIC00B using the PICkit 2 Programmer application without any AC-DC adapters.

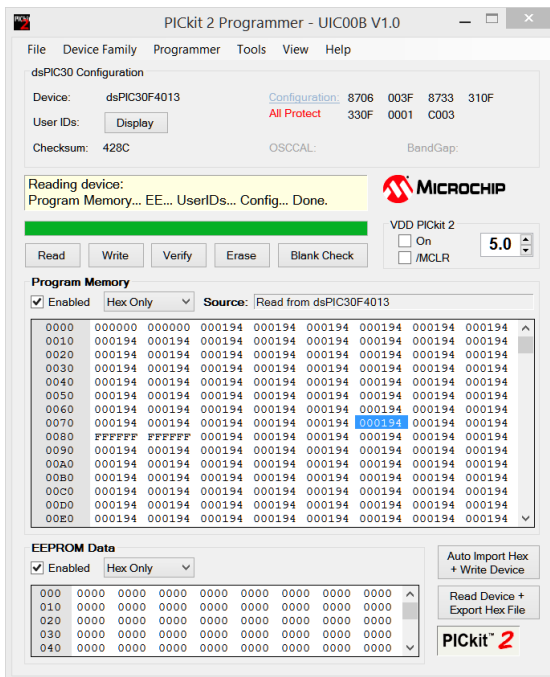


Figure 3.6. Overview of PICkit 2 software user interface

3.2.4 Oscilloscope

Due to many implemented sensors provide analog input, the handheld oscilloscope from Pico Technologies, PicoScope is used to analyze the inputs. The PicoScope 6 combines the function of oscilloscope, spectrum analyzer and waveform generator into one. It is easy to operate and it powers itself with the USB connection from PC.

The software of PicoScope is free to download and it provides many features, some of the main features and functions are listed below.

- Oscilloscope and XY oscilloscope which capable of plotting one channel again another
- Spectrum analyzer for analyzing and viewing signals in different frequency domain
- Arbitrary waveform generator in any shape
- Digital persistence which allows to spot glitches
- Math channels for configuring channels, or create own functions
- Automated measurements and statistics display
- Mask limit testing for observing the waveform
- Serial decoding from a serial bus such as CAN bus

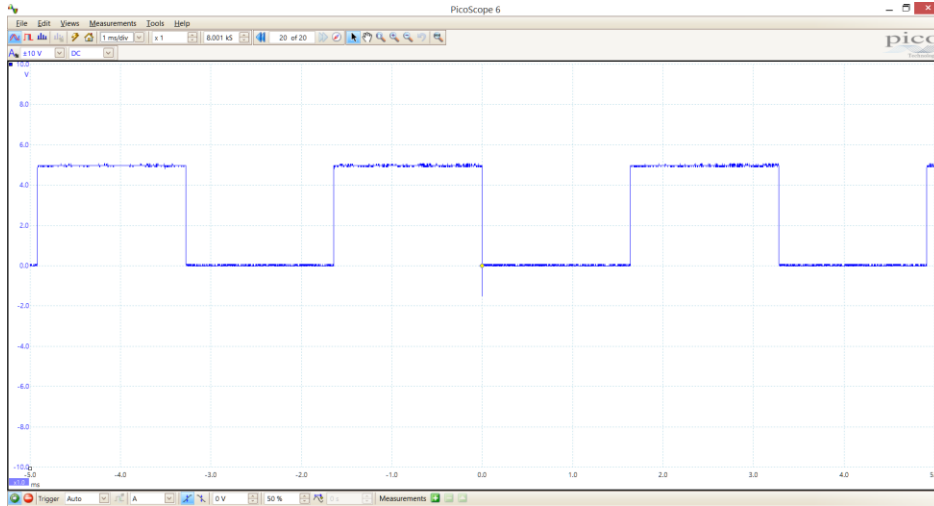


Figure 3.7. Overview of PicoScope 6 software user interface



Figure 3.8. Overview of the PicoScope handheld oscilloscope

3.3 Product Components

3.3.1 Center Processing Unit

SKds40A is a microcontroller startup kit specially designed for a few type of microcontroller from the Microchip 40-pin dsPIC30F microcontroller family. It supports dsPIC30F3011, dsPIC30F4011, dsPIC30F3014, and dsPIC30F4013 microcontroller. It is a board with basic components for beginning project development but all program and interface should be developed by the user.

The SKds40A is chosen along with a dsPIC30F4013 microcontroller to be the center processing unit of this project's robot. It offers plug and use features, and many else:

- Can be powered by 7V to 15V AC-DC Adaptor
- Can be directly connected by UIC00B programmer for programming or debugging
- Contains dedicated pins and slot for a parallel 2x16 LCD with supported backlight
- Contains dedicated pins and slot for UC00A (USB to UART Converter)
- Pad with jumpers to select different UART mode
- Socket for swappable external crystal oscillator, default 10MHz crystal is used
- 2 x LED indicator
- 2 x Programmable switch

The Microchip dsPIC30F4013 microcontroller is one of the Microchip 16-bit embedded control solutions which contains extensive Digital Signal Processor (DSP) functionality

within its 16-bit high performance microcontroller architecture. It offers many advance features despite its low cost and small size, some of the features are listed below:

- High Performance modified RISC CPU Features Overview:
 - Up to 30MIPS Operation
 - Up to 33 Interrupt Sources
 - Modified Harvard Architecture
 - C Compiler Optimized Instruction Set Architecture
 - 48 KB On-Chip Flash Program Space
 - 2 KB of On-Chip Data RAM
 - 1 KB of nonvolatile Data EEPROM
 - 16 x 16-bit Working Register Array
- Peripheral Features Overview:
 - Up to 13 input channels for 12-bit Analog-to-Digital Converter (ADC) with 200 ksps conversion rate
 - Up to 5 16-bit Timers/Counters
 - Up to 4 16-bit Capture Input Functions
 - Up to 4 16-bit Compare/PWM Output Functions
 - Up to 2 UART modules with FIFO buffers
 - Data Converter Interface (DCI) for common Audio Codec Protocols
 - 3-Wire Serial Peripheral Interface (SPI) Module
 - I²C module with Multi-Master/Slave mode
 - CAN bus module with CAN 2.0B Standard
- Digital Signal Processor and Special Microcontroller Features
- CMOS Technology



Figure 3.9. Overview of SKds40A attached with a dsPIC30F4013 and a 2x16 LCD

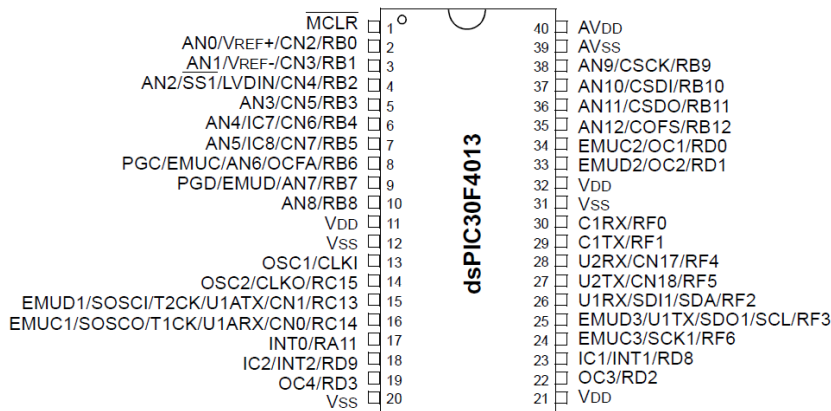


Figure 3.10. Pin Diagram of the dsPIC30F4013 microcontroller

3.3.2 Motor controller

To allow the robot move in different direction and various speeds, the L298N Dual H Bridge Motor Driver Controller module board is implemented to control the 4 servomotors attached with rubber wheel each.

The board receives 2 external power inputs and supports 4 power outputs. The board size is 55mm x 49mm x 33mm with a fixed copper pillar and a heat sink. It can be driven by +5 V to +35 V power.



Figure 3.11. Overview of L298N Dual H Bridge Motor Driver Controller module

3.3.3 Position detection sensor

To calculate the position and movement of the robot, the SN-IMU5D-LC is implemented to provide analog inputs of the related information. The SN-IMU5D-LC is a low cost 5 DOF Inertial Measurement Unit which incorporates a 3-Axis Accelerometer and 2 x Gyro Sensor. A low-noise amplifier with 4.7x gain is included in the module to increase to sensitivity of the Gyro sensors and provides extra output.



Figure 3.12. Overview of SN-IMU5D-LC

The IMU can be powered by +5V with onboard 3.3V voltage regulator. The 5-DOF analog data output from the IMU are pitch and roll rate for the Gyro sensors, and X, Y and Z Axis for the Accelerometer. The dimension of this unit is 41.91mm x 17.94mm.

The Accelerometer in the board is the Analog Devices ADXL335 3-Axis Accelerometer. It is low power, small and thin for only 4mm x 4mm x 1.45mm, and with signal conditioned voltage outputs. It measures acceleration with a minimum range of +-3g and survives shock up to 10,000 g. It can also measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from vibration, shock, or motion.

The Gyro sensors onboard are the Murata ENC-03R. This sensor is an angular rate sensor that uses the Coriolis force phenomenon, which is generated when a rotational angular rate is applied to the vibrator. It is ultra-small and ultra-lightweight, under 0.1cc, but provides quick responses, consume low current and low driving voltage.

However, the output of the Gyro sensors are not yet implement in this project due to limited applications but will be considered in the future development.

3.3.4 Range detection sensor

Currently, only 1 range sensor is implemented in this project. The LV-MaxSonar-EZ1 High Performance Sonar Range Finder is a small and light weight module offered by Maxbotix Inc. It can be supplied by 2.5V to 5.5V power and provides 3 different types of output for the readings and many more features:

- Continuously variable gain for beam control and side lobe suppression
- Object detection down to zero range objects and up to 6.45 meters
- Fast rate of readings which can occur up to every 50 milliseconds
- Analog interface, pulse width interface, and serial interface which are active simultaneously
- Very low cost but provides reliable and stable range data
- Very low power consumption which are good for multiple sensors or battery based systems.
- Narrow beam width with side lobe suppression



Figure 3.13. Overview of LV-MaxSonar-EZ1 High Performance Sonar Range Finder

3.3.5 Radio Frequency (RF) communication

Out of many types of wireless communication, the 2.4GHz band is chosen. The implemented hardware is the latest 2.4GHz wireless transceiver module from Nordic semiconductor, the nRF24L01+.

The nRF24L01+ is a single chip 2.4GHz wireless transceiver with an embedded baseband protocol engine specially designed for ultra-low power wireless applications. It is designed for operation in the world wide ISM frequency band at 2.400 to 2.4835GHz.

The nRF24L01+ has to be configured and operated through a Serial Peripheral Interface (SPI) for its register map which contains all configuration register and is accessible in all operation modes of the chip.

The embedded baseband protocol engine (Enhanced ShockBurst) is based on packet communication and supports various modes from manual operation to advance autonomous protocol operation.

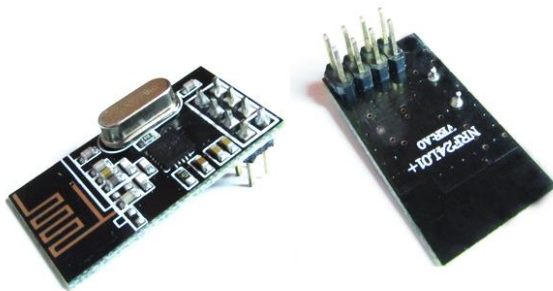


Figure 3.14. Overview of the nRF24L01+ module

The dimension of the module is only 15mm x 29mm. Two of the nRF20L01+ are used in this project, 1 implemented in the robot and the other implemented in the controller side.

More detailed features of the nRF24L01+ provided by the manufacture includes:

- Radio
 - 126 RF channels
 - GFSK modulation
 - 1 and 2 Mbps air data rate
- Transmitter
 - Programmable output power: 0, -6, -12 or -18dBm
- Receiver
 - Programmable LNA gain
 - Integrated channel filters
- RF Synthesizer
 - Fully integrated synthesizer without external loop filter
- Enhanced ShockBurst
 - 1 to 32 bytes dynamic payload length
 - Automatic packet handling
- Power Management
 - Integrated voltage regulator
 - 1.9 to 3.6V supply range
- Host Interface
 - 4-pin hardware SPI
 - 5V tolerant inputs

CHAPTER 4. METHODOLOGY AND APPLICATION

Upon power up, the robot will be in manual mode by default. The user will have to select between autonomous mode and manual mode using the wireless communication device provided.

In manual mode, the user can instruct the robot to move, stop or turn, in various preset speed. In this mode, range finder and positioning sensors will be deactivated. The robot will be set to stop moving by default when change into manual mode.

In autonomous mode, the robot will start to move itself first by forwarding. The ultrasonic sensor will start detecting obstacles in front of it. The IMU will also provide information of the position of the robot.

If the robot detects obstacles in front of it exceed a minimum length (refer to Ultrasonic section), it will rotate itself counterclockwise until the ultrasonic sensor detects nothing else and continue to forward.

If the robot is going downhill or sloppy area and the Y-axis of the robot position reach a certain limit (refer to the IMU section), the robot will slow down its speed. If the robot is going uphill and the Y-axis of the robot position reach a certain limit (refer to IMU section), the robot will reverse for a while and change another direction to prevent a backflip.

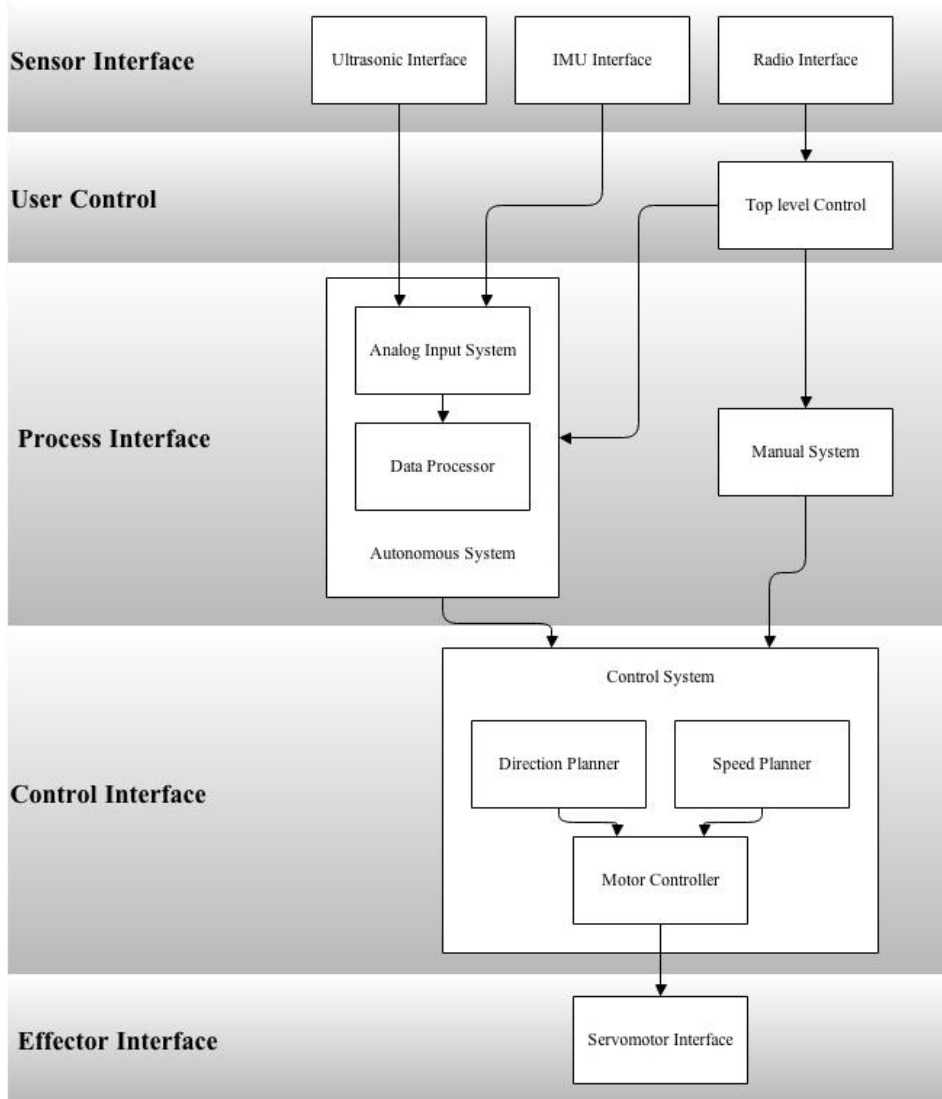


Figure 4.1. Operation flow of the robot

4.1 Analog-to-Digital Converter

Initially, all ports of the dsPIC30F4013 are set as digital outputs. According to the dsPIC30F4013 datasheet, setting up the ADC requires several steps in configuring the ADC register map.

All ADC ports are mapped to the Port B of the dsPIC30F4013 and there are total 13 pins. Due to RB0 to RB7 are mapped to the 2x16 LCD by default, with a RB8 unattended in the same row, RB9 to RB12 are chosen to read the analog inputs from the sensors.

The function `adc_init()` initializes and configures the ADC features in the dsPIC30F4013 registry. In the function, the Data Direction register (TRISx) B is set to 0x1E00 (0001 1110 0000 0000 in binary), every bit here (until bit 12) corresponds to one of the ports; Setting a bit to '1' configures the pin to be output, and clearing a bit to '0' means an input. Bit 0 starting from the right, bit 9, 10, 11 and 12 are set to '1' which means these 4 pins are outputs.

```
TRISB = 0x1E00;
```

For AD Port Configuration register (ADPCFG), it determines whether a pin is a digital input by setting '1', and is an analog input by clearing to '0'. Setting to 0x01FF (0000 0001 1111 1111 in binary) configures pin RB9, RB10, RB11, and RB12 to be analog input.

```
ADPCFG = 0x01FF;
```

The AD Channel Select register (ADCHS) is set to all zero because we will be using auto channel scanning mode so that we do not need to select the channel manually. However,

the AD Input Scan Selection register (ADCSSL) which selects input for scanning are set to 0x1E00 (0001 1110 0000 0000 in binary) because the 4 pins B9, RB10, RB11 and RB12 will be selected for the auto channel scanning mode.

`ADCHS = 0x0000;`

`ADCSSL = 0x1E00;`

The AD Control registers (ADCONx) controls different operations of the AD module. ADCON1 is set to 0x00E4 (0000 0000 1110 0100 in binary) where bit 5, 6 and 7 (SSRC<2:0>) which enables the Auto-Convert mode for auto-converting the analog inputs, and bit 2 (ASAM) which enables automatic sampling, are set to '1'.

`ADCON1 = 0x00E4;`

The ADCON2 is set to 0x040C (0000 0100 0000 1100 in binary) where bit 10 (CSCNA) which enables the channel scanning mode, and bit 2 and 3 (SMPI<3:0>) which selects the number of conversion sequences that would be performed before an interrupt occurs (0011 selects 3 conversion sequences), are set to '1'.

`ADCON2 = 0x040C;`

The ADCON3 is set to 0x0A3F (0000 1010 0011 1111 in binary). Bit 0 to 5 (ADCS<5:0>) which determines the sampling speed are set to all '1' (6 bits with maximum value of 63), thus the dsPIC30F4013 will be running at 30 MIPS. Bit 8 to bit 12 (SAMC<4:0>) which selects the number of AD clocks between the start of acquisition and conversion, are set to 01010 (5 bits with value of 10).

Bit 13 to 15 (VCFG<2:0>) which sets the reference voltages sources for the input, is set to

0x000 makes the Upper reference voltage to be the analog positive power rail (5V) and the Lower reference voltage to be the analog negative power rail (Ground), which means the analog input will be range from 0 to 5120.

```
ADCON3 = 0x0A3F;
```

The bit 15 of ADCON1 (ADON) enables the ADC by setting to '1'. This configuration is usually to be done after configuring others and this is the last configuration in the ADC initializing function.

```
ADCON1bits.ADON = 1;
```

The bit 0 of ADCON1 (DONE) will automatically set to '1' after a cycle of sampling and conversion is done. The data will be stored into ADCBUFx accordingly. ADCBUF0 is the first data buffer and it will store the first data obtained from the first enabled channel.

Due to all the analog input range from 0 to 5120, for better calculation in 12-bit, the input will be divided by 5 when storing from the ADCBUFx to a variable. For a much stable readings without sudden big noises, the variable will get the average of 10 analog readings after having saved every 10 of them using a counter.

CHAPTER 4. METHODOLOGY AND APPLICATION

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISA	02C0	—	—	—	—	TRISA11	—	—	—	—	—	—	—	—	—	—	—	0000 1000 0000 0000
PORTA	02C2	—	—	—	—	RA11	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	LATA11	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0001 1111 1111 1111
PORTB	02C8	—	—	—	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	—	—	—	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	—	—	—	—	1110 0000 0000 0000
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	TRISD9	TRISD8	—	—	—	—	—	TRISD3	TRISD2	TRISD1	TRISD0	0000 0011 0000 1111
PORTD	02D4	—	—	—	—	—	RD9	RD8	—	—	—	—	—	RD3	RD2	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	LATD9	LATD8	—	—	—	—	LATD3	LATD2	LATD1	LATD0	—	0000 0000 0000 0000
TRISF	02DE	—	—	—	—	—	—	—	—	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	—	0000 0000 0111 1111
PORTF	02E0	—	—	—	—	—	—	—	—	RF6	RF5	RF4	RF3	RF2	RF1	RF0	—	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	—	0000 0000 0000 0000

Table 4.1. dsPIC30F4013 Port Register Map (taken from dsPIC30F4013 Data Sheet)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
ADCBUF0	0280	—	—	—	—	ADC Data Buffer 0											0000 uuuu uuuu uuuu	
ADCBUF1	0282	—	—	—	—	ADC Data Buffer 1											0000 uuuu uuuu uuuu	
ADCBUF2	0284	—	—	—	—	ADC Data Buffer 2											0000 uuuu uuuu uuuu	
ADCBUF3	0286	—	—	—	—	ADC Data Buffer 3											0000 uuuu uuuu uuuu	
ADCBUF4	0288	—	—	—	—	ADC Data Buffer 4											0000 uuuu uuuu uuuu	
ADCBUF5	028A	—	—	—	—	ADC Data Buffer 5											0000 uuuu uuuu uuuu	
ADCBUF6	028C	—	—	—	—	ADC Data Buffer 6											0000 uuuu uuuu uuuu	
ADCBUF7	028E	—	—	—	—	ADC Data Buffer 7											0000 uuuu uuuu uuuu	
ADCBUF8	0290	—	—	—	—	ADC Data Buffer 8											0000 uuuu uuuu uuuu	
ADCBUF9	0292	—	—	—	—	ADC Data Buffer 9											0000 uuuu uuuu uuuu	
ADCBUFA	0294	—	—	—	—	ADC Data Buffer 10											0000 uuuu uuuu uuuu	
ADCBUFB	0296	—	—	—	—	ADC Data Buffer 11											0000 uuuu uuuu uuuu	
ADCBUFC	0298	—	—	—	—	ADC Data Buffer 12											0000 uuuu uuuu uuuu	
ADCBUFD	029A	—	—	—	—	ADC Data Buffer 13											0000 uuuu uuuu uuuu	
ADCBUFE	029C	—	—	—	—	ADC Data Buffer 14											0000 uuuu uuuu uuuu	
ADCBUFF	029E	—	—	—	—	ADC Data Buffer 15											0000 uuuu uuuu uuuu	
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>	—	—	SSRC<2:0>	—	—	ASAM	SAMP	DONE	—	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	BUFS	—	SMPI<3:0>			—	—	—	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>				ADRC	—	ADCS<5:0>			—				0000 0000 0000 0000
ADCHS	02A6	—	—	—	CH0NB	CH0SB<3:0>				—	—	—	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

Table 4.2. dsPIC30F4013 ADC Register Map (taken from dsPIC30F4013 Data Sheet)

4.2.1 Ultrasonic Range Finder

Out of the 3 different types of outputs from the Ultrasonic sensor, the Analog output from the sensor is chosen for this project. The AN pin of the Ultrasonic sensor is connected to RB9 of the dsPIC30F4013.

According to the datasheet from the manufacturer, the sensor outputs analog voltage with a scaling factor of $(V_{cc}/512)$ per inch (1 inch: 2.54cm). As the power supply to the sensor is 5V, the output will be about 9.8mV/in (3.8mV/cm). After converting by the dsPIC30F4013, from zero-range to 20cm will read a value of 8 out of 1024, a 30cm range directly in front of the sensor will read a value of 15 out of 1024.

The chosen value to trigger a direction changing event is 15 (30cm), which the robot will rotate itself counter-clockwise (or clockwise according to configuration) when the analog reading from Ultrasonic sensor gives a value of bigger than 15 out of 1024. All values are nominal and in testing.

4.2.2 Inertial Measurement Unit

The IMU module provides 7 analog outputs. X, Y and Z from Accelerometer, X and Y from Gyro sensors, and another 4.7x amplified X and Y from Gyro sensors. Currently, only the 3 outputs from the Accelerometer is in used.

The reading input processing and event triggering value calculation method is different from the other sensors. The default values are taken directly from the processed analog reading. For example, when the robot is placed horizontally on a smooth surface, the reading value of Y-axis is 265. When the robot is placed vertically facing top (the sky) in 90 degree perpendicular to the horizon surface (ground), the reading value of Y-axis is 325. The calculation applies to X-axis too.

According to the data obtained from the experiments above, for every degree the robot rotates, the reading value increases or decrease by 0.666. The maximum changes of the value is set at 90 degree, 0 degree and 180 degree give the same value.

Currently, only Y-axis is implemented. When the robot is going uphill and reached nearly 67 degree to the top (reading value is around 310), it will reverse and rotate to another direction to prevent back flipping. When going downhill and reached nearly 45 degree to the bottom (reading value is around 235), the robot will slow down its speed. All values are nominal and in testing.

4.3 Pulse Wave Modulation

The dsPIC30F4013 can generate PWM using its Output Compare Module. One of the module's features is the Simple PWM mode which is much easier to configure. There are total 4 pins in the microcontroller which support the module: OC1 (RD0), OC2 (RD1), OC3 (RD2) and OC4 (RD3). Only OC1 and OC2 are currently implemented.

The initialization of the Pulse Wave Modulation for this project starts with setting the Period Register 1 (PR1) in the dsPIC30F4013 Timer module 16-bit Timer Mode to 0x00FF (255 in decimal). The value of PR1 corresponds to the desired elapsed time between interrupts. The Output Compare module uses Timer 2.

```
PR1 = 0x00FF;
```

Next, the Output Compare Second Register (OCxRS, x represents the pin number, 1, 2, 3 or 4) which is the duty cycle is initialized to 0xFFFF. Then the Output Compare Main Register (OCxR) which is the initial duty cycle for the output is set to 0x0000. These settings create a full output power.

The OCxRS can be changed to any value between 0x0000 and 0xFFFF to create different duty cycles. For example, 0x7FFF creates a 50% duty cycle which outputs half of the full power.

```
OC1RS = 0x0000;
```

```
OC2RS = 0x0000;
```

```
OC1R = 0xFFFF;
```

OC2R = 0xFFFF;

Then, Output Compare Configuration register (OCxCON) is set to 0x0006 (0000 0000 0000 0110 in binary), hence the bit 0 to 2 (OCM<2:0>) is set to 110, which enables Simple PWM mode for the selected pin.

OC1CON = 0x0006;

OC2CON = 0x0006;

Lastly, the Timer 2 Configuration register (T2CON) is set to 0x8000 (1000 0000 0000 0000 in binary). The bit 15 (TON) is set to '1' to enable the Timer 2 which also starts the Output Compare Module Simple PWM Mode.

T2CON = 0x8000;

Currently, there are 2 preset speed for the robot which is 100% speed (0xFFFF) and 50% speed (0x7FFF).

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
OC1RS	0180	Output Compare 1 Secondary Register																0000 0000 0000 0000
OC1R	0182	Output Compare 1 Main Register																0000 0000 0000 0000
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000 0000 0000 0000
OC2RS	0186	Output Compare 2 Secondary Register																0000 0000 0000 0000
OC2R	0188	Output Compare 2 Main Register																0000 0000 0000 0000
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSE	OCM<2:0>			0000 0000 0000 0000
OC3RS	018C	Output Compare 3 Secondary Register																0000 0000 0000 0000
OC3R	018E	Output Compare 3 Main Register																0000 0000 0000 0000
OC3CON	0190	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000 0000 0000 0000
OC4RS	0192	Output Compare 4 Secondary Register																0000 0000 0000 0000
OC4R	0194	Output Compare 4 Main Register																0000 0000 0000 0000
OC4CON	0196	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000 0000 0000 0000

Table 4.3. dsPIC30F4013 Output Compare Register Map (taken from dsPIC30F4013 Data Sheet)

4.3.1 Servomotors

The L298N Dual H Bridge Motor Driver Controller module has 4 outputs: A, B, C and D. The 4 outputs are categories into Left and Right. Two outputs for Left forward (Output A, defined as L1) and Left reverse (Output B, defined as L0), and another two outputs for Right forward (Output C, defined as R1) and Right reverse (Output D, defined as R0). These 4 outputs are wired to the 8 pins from the 4 servomotors accordingly.

The module receives 4 inputs to trigger its 4 outputs to the servomotors, and 2 inputs for the speed (PWM): ENA which controls the speed of Left outputs: A and B, and ENB which controls the speed of Right outputs: C and D.

Four pins of the dsPIC30F4013: RF5, RC14, RF4 and RD9 are connected to the A (L1), B (L0), C (R1) and D (R0) of the L298N Dual H Bride motor driver controller module. And OC1 (RD0) and OC2 (RD1) of dsPIC30F4013 are connected to the ENA and ENB of the module.

The 4 digital output pins of the dsPIC30F4013 determines the moving and direction of the robot. Only maximum 2 out of the 4 pins can be set to '1' in every operation. To move forward, L1 and R1 are set to '1'. To rotate in a clockwise direction, L1 and R0 is set to '1', which the Left side are moving forward and the Right side are moving reverse. To stop moving, either all digital pins are clear to '0' or the PWM output pins are set to 0x0000.

4.4 RF Transceiver

The nFR24L01+ has 8 pins: 2 power (3.3V and Ground), 2 digital outputs (output to the microcontroller) and 4 digital input pins.

One of the 2 output is the Interrupt Request (IRQ) which is a maskable interrupt pin that output '1' when waiting for interruption and output '0' when there is an interrupt (incoming data). The other output is the Master-In-Slave-Out (MISO) which outputs the received data to the microcontroller, this pin is connected to SPI Input pin (SDI1, or RF2) of the dsPIC30F4013.

One of the digital input pin, Master-Out-Slave-In (MOSI) reads the to-be-sent data from the microcontroller and sends out to another module. It is connected to SPI Output pin (SDO, or RF3) of the dsPIC30F4013.

The remaining 3 input pin, Chip Enable (CE) which disables the current mode (Transmit Tx mode or Receive Rx mode) when receives '0' from the microcontroller, and reactivate to another mode after modifying the buffer when receives '1' from the microcontroller. SPI Chip Select NOT (CSN) is always on '1', any SPI operations will only start after the CSN is set to '0'. The SCK pin reads the clock rate from the microcontroller.

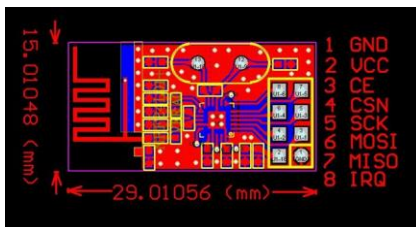


Figure 4.2. Diagram of the nFR24L01+

In the initialization of the SPI, the SPI Status register (SPI1STAT) bit 6 SPI Overflow bit (SPIROV) is cleared to '0' which indicates there is no buffer overflow condition. Then SPI1STAT bit 15 SPI Enable (SPIEN) is cleared to '0' to temporary disable the SPI feature in order to perform the following configurations without errors.

`SPI1STATbits.SPIROV = 0;`

`SPI1STATbits.SPIEN = 0;`

Then SPI Configuration register (SPI1CON) is set to 0x0320 (0000 0011 0010 0000 in binary) where bit 5 (MSTEN) which enables master mode, bit 8 (CKE) and bit 9 (SMP) which determines whether transmit occurs on transition from active clock state to idle clock state, is set to '1'.

Bit 6 (CKP) which selects the idle clock state (high or low) and bit 10 (MODE16) which enables 16-bit mode from 18-bit mode are cleared to '0'. Bit 2, 3 and 4 (SPRE0, SPRE1, SPRE2) are cleared to '0' too for the slowest SPI clock rate because the nRF24L01+ compatible speed is slow. Lastly, the SPIEN is set to '1' to enable the SPI features again.

`SPI1CON = 0x0320;`

`SPI1STATbits.SPIEN = 1;`

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
SPI1STAT	0220	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	Transmit and Receive Buffer																0000 0000 0000 0000

Table 4.4. dsPIC30F4013 Serial Peripheral Interface Register Map (taken from dsPIC30F4013 Data Sheet)

The initialization of the nRF24L01+ module starts with flushing its Transmit pin (Tx) and Receive pin (Rx) using the preset commands from the library provided by the manufacturer. There are few preset functions, commands and address provided in the library. An array of 5 unsigned character (aucBuffer) is created to store temporary value to be written.

```
unsigned char aucBuffer[5];
```

The void function vWriteNrf24l01 (unsigned char, unsigned char, const unsigned char*, unsigned char) takes in 4 arguments which is the preset commands, register address to write, buffer for data, and the length of the data in bytes.

```
vWriteNrf24l01(FLUSH_TX, 0, 0, 0);
```

```
vWriteNrf24l01(FLUSH_RX, 0, 0, 0);
```

Then the physical address of the module is written into the aucBuffer and store into the register. Both Tx and Rx are configured to have the same physical address because both the nRF24L01+ modules (1 for robot, the other for controller) are using the same code.

```
aucBuffer[0] = 0xaa;
```

```
aucBuffer[1] = 0xbb;
```

```
aucBuffer[2] = 0xcc;
```

```
aucBuffer[3] = 0xdd;
```

```
aucBuffer[4] = 0xee;
```

```
vWriteNrf24l01(W_REGISTER, ADD_TX_ADDR, &aucBuffer, 5);
```

```
vWriteNrf24l01(W_REGISTER, ADD_RX_ADDR_P0, &aucBuffer, 5);
```

After that, 1 of the 6 parallel data pipes is configured for auto acknowledgement mode (ACK). The data pipe used is Pipe 0. It is enabled after the ACK configuration is done.

```
aucBuffer[0] = 0x01;
```

```
vWriteNrf24l01(W_REGISTER, ADD_EN_AA, &aucBuffer, 1);
```

```
aucBuffer[0] = 0x01;
```

```
vWriteNrf24l01(W_REGISTER, ADD_EN_RXADDR, &aucBuffer, 1);
```

Continued by configuring the Rx payload size to 1 byte. The value for the predefined variable (DATA_LENGTH) is '1'.

```
aucBuffer[0] = DATA_LENGTH;
```

```
vWriteNrf24l01(W_REGISTER, ADD_RX_PW_P0, &aucBuffer, 1);
```

Next is setting up the auto retransmission delay and count for the Tx. The aucBuffer is set to 0b00011111 (binary) where bit 0 to 3 (ARC, Auto Retransmit Count) is '1111' (15 in decimal) which sets the auto retransmission count to 15, and bit 4 to 7 (ARD, Auto Retransmit Delay) is '0001' which sets the auto retransmission delay to 500uS according to the nRF24L01+ data sheet provided.

```
aucBuffer[0] = 0b00011111;
```

```
vWriteNrf24l01(W_REGISTER, ADD_SETUP_RETR, &aucBuffer, 1);
```

Then, the RF channel is set to channel 50. The value for the predefined variable (RF_CHANNEL) is '50'.

```
aucBuffer[0] = RF_CHANNEL;
```

```
vWriteNrf24l01(W_REGISTER, ADD_RF_CH, &aucBuffer, 1);
```

Next is setting the air data rate and the power of transmission. The aucBuffer is set to 0b00000110 (binary) where bit 1 and 2 (RF_PWR) is set to '11' which configures the power of transmission to 0dBm according to the datasheet ('00' for -18dBm, -6dBm step). Bit 3 (RF_DR) that configure the data rate is set to '0', which is 1Mbps ('1' for 2Mbps).

```
aucBuffer[0] = 0b00000110; // 1Mbps.
```

```
vWriteNrf24l01(W_REGISTER, ADD_RF_SETUP, &aucBuffer, 1);
```

Finally, the ADD_CONFIG is configured with the aucBuffer set to 0b00001111. Bit 0 (PRIM_RX) which sets the primary control as Rx, bit 1 (PWR_UP) which powers up the control, bit 2 (CRCO) which sets the CRC encoding scheme ('0' for 1 bytes, '1' for 2 bytes), and bit 3 (EN_CRC) which enables the CRC when set to '1', is all set to '1'.

```
aucBuffer[0] = 0b00001111;
```

```
vWriteNrf24l01(W_REGISTER, ADD_CONFIG, &aucBuffer, 1);
```

Lastly, the Receive mode is enabled by setting the Chip Enable bit (CE) to '1' and the initialization function of the nRF24L01+ module is completed. The Register Map of nRF24L01+ can be obtained from its Data Sheet provided from the manufacturer.

4.4.1 RF Implementation

When using the nRF24L01+, the algorithms and functions of the Receive mode are to be processed separately from the Transmit mode. The default status of the nRF24L01+ is in Received mode.

Start with the Interrupt Request pin (IRQ), if there is data received from the other RF module, the IRQ will be cleared to '0'. Then the preset function `ucNrf24l01ReceivePacket (unsigned char *, unsigned char)` will return '1' if the data is successfully received, read and stored into the first parameter.

At this point the data received can be further processed in a function or an algorithms. For example, the robot will stop moving when the received data is '0', and change direction when the received data is '5'. If there are many different receivable data, a switch case is recommended. The variable `ucStatus` and `ucData` are self-created.

```

if (IRQ == 0) {
    ucStatus = ucNrf24l01ReceivePacket(&ucData, 1);
    if (ucStatus != 0) {
        switch (ucData) {
            case 1:
        }
    }
}

```

For transmitting data to the other RF module, Transmit mode has to be enabled first by running the preset function `vNrf24l01EnableTx ()` from the library. Then use another preset function `ucNrf24l01TransmitPacket (unsigned char *, unsigned char)` to transmit the custom data by putting it into the first parameter.

The function will return '0' if the transmission is failed. Lastly, Receive mode has to be re-enabled by running the preset function `vNrf24l01EnableRx ()`. The variable `ucStatus` and `ucData` are self-created.

```
vNrf24l01EnableTx();
```

```
ucData = 1;
```

```
ucStatus = ucNrf24l01TransmitPacket(&ucData, 1);
```

```
vNrf24l01EnableRx();
```

4.5 Information Output

A 16 x 2 LCD is attached to the SKds40A board can be used to provide on the spot sensor readings. The LCD uses a default library from the manufacturer and it uses pin RB0 to RB7 of the dsPIC30F4013. It can display data sent from the dsPIC30F4013 but the data has to be in String format.

The UART module is also ready to be implemented anytime through a UIC00A Serial to USB Converter using the default library from its manufacturer and the data sent from the dsPIC30F4013 has to be in String format too.

Therefore, all the readings (mostly analog readings) will be combined into a String using the function `sprintf ()` before connecting with the LCD or the UART.

CHAPTER 5. PROJECT REVIEW

5.1 Project Discussion and Conclusion

This paper presents the details and design of the semi-autonomous robot of the project. The current form of the robot is able to read data from its sensors and perform further operations such as slowing down movement speed and changing directions.

The ultrasonic sensor offers great reading speed and stability. However, the detection area is only the linear front side of the sensor instead of a more than 90 degree circular sector. The other area outside the linear area in front of the sensor are its blind spots. Hence some small obstacles which is not in the linear detection area of the sensor are not detected.

In order to get an accurate reflection of the ultrasonic projectile, the obstacle to be detected has to be facing the sensor which both side is to be parallel. Sometimes the problems above will cause the robot to get in touch with some obstacles.

The dsPIC30F4013 works well with the motor controller and the servomotors. The robot can be left to move autonomously in a normal area full of noticeable obstacles for a long period.

The RF communication can be used to correct the robot's position or direction when the robot encountered some unsolvable case. For example, when the robot get in touch with an obstacle which the ultrasonic sensor could not detect, the user can manually control the robot to change its direction.

5.2 Future Work

This project offers many opportunities for future improvement and development for the robot prototype. More sensors can be implemented in different part of the robot to decrease the blind spot of the obstacle detection sensors.

With better components and peripherals allocation, the design platform on top of the robot chassis will provide more spaces for further add-ons and development. For example, a robot arm or an image detection sensor can be implemented onto the robot. The robot components can be modified or changed according to the implementation field and serving area.

The current total cost of all components of the robot is not appraisable as most of them still serves wide range of features instead of a custom designated function. This problem can be improved in the future with better planning and design.

BIBLIOGRAPHY

Department of Physics and Electronic Engineering, University of Waikato (2004) *A Human-Like Semi Autonomous Mobile Security Robot*. [report] Wellington, New Zealand: Robotechnology.

En.wikipedia.org (2002) *Autonomous robot - Wikipedia, the free encyclopedia*. [online] Available at: http://en.wikipedia.org/wiki/Autonomous_robot [Accessed: 7 Apr 2013].

Russell, S. and Norvig, P. (2010) *Artificial Intelligence: A Modern Approach*. 3rd ed. PEARSON, p.971.

Sawicz, D. (n.d.) *Hobby Servo*. [e-book] <http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf> [Accessed: 07th April 2013].

Nordic Semiconductor. (n.d.) *nRF24L01 Single Chip 2.4GHz Transceiver*. [e-book] Nordic Semiconductor. Available through: Cytron Technologies http://www.cytron.com.my/usr_attachment/nRF24L01_Product_Specification_v2_0.pdf [Accessed: 07 April 2013].

Writer, C. (2012) *How Do IR Sensors Work?*. [online] Available at: http://www.ehow.com/how-does_5561845_do-ir-sensors-work.html [Accessed: 7 Apr 2013].

Unknown. (2012) *Emlabz Product Catalogue*. [e-book] Emlabz. p.10. Available through: Emlabz Technology.

BIOBLOGRAPHY

Angular Rate Sensors ENC-03R. n.d. [e-book] Murata. Available through: Microchip Technologies <http://www.cytron.com.my/viewProduct.php?pcode=RF-TRX-2.4G&name=RF%20Transceiver%202.4G> [Accessed: 10 Apr 2014].

Burke, T. 2012. dsPIC30F4011 Super Example. *ad hocumentation*, [blog] 2012, Available at: http://batchloaf.wordpress.com/dspic30f4011_super_example/ [Accessed: 01 Apr 2014].

dsPIC30F3014/4013 Data Sheet. 2010. [e-book] Microchip Technology Inc. Available through: Microchip Technology [Accessed: 01 Apr 2014].

Huddleston, C. 2007. *Intelligent Sensor Design Using the Microchip dsPIC*. [e-book] USA: Newnes. [Accessed: 01 Apr 2014].

L298N. 2000. [e-book] Italy: STMicroelectronics. www.st.com [Accessed: 10 Apr 2014].

LV-MaxSonar-EZ1 High Performance Sonar Range Finder. 2012. [e-book] Maxtibox Inc. [Accessed: 10 Apr 2014].

Microchip.com. 2014. *Development Tools*. [online] Available at: <http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DV164121> [Accessed: 10 Apr 2014].

Microchip.com. 2014. *MPLABX / Microchip Technology Inc.*. [online] Available at: <https://www.microchip.com/pagehandler/en-us/family/mplabx/> [Accessed: 10 Apr 2014].

BIBLIOGRAPHY

- Mikroe.com. n.d. *MikroElektronika - Development tools, Compilers, Books*. [online] Available at: <http://www.mikroe.com/chapters/view/54/chapter-6-output-compare-module/#ch6.3> [Accessed: 01 Apr 2014].
- nRF24L01 tutorials 1-3 for the PIC completed. 2007. *DIY embedded*, [blog] 2007, Available at: <http://blog.diyembedded.com/2007/06/tutorials-1-3-for-pic-completed.html> [Accessed: 10 Apr 2014].
- Picotech.com. 2014. *PicoScope Oscilloscope Software the standard in PC Oscilloscope software*. [online] Available at: <http://www.picotech.com/picoscope-oscilloscope-software.html> [Accessed: Apr 2014].
- Ruangpayoongsak, N., Roth, H. and Chudoba, J. 2005. *Mobile Robots for Search and Rescue*. [report] Kobe, Japan: International Workshop on Safety, Security and Rescue Robotics.
- Small, Low Power, 3-Axis +/-3 g Accelerometer ADXL335. 2009. [e-book] Analog Devices. Available through: Cytron Technologies <http://www.cytron.com.my/viewProduct.php?pcode=RF-TRX-2.4G&name=RF%20Transceiver%202.4G> [Accessed: 10 Apr 2014].

APPENDICES A. REGISTER MAP OF NRF24L01+ FROM DATA SHEET

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0:POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control 1: PRX, 0: PTX
01	EN_AA Enhanced ShockBurst™				Enable 'Auto Acknowledgment' Function Disable this functionality to be compatible with nRF2401.
	Reserved	7:6	00	R/W	Only '00' allowed
	ENAA_P5	5	1	R/W	Enable auto acknowledgement data pipe 5
	ENAA_P4	4	1	R/W	Enable auto acknowledgement data pipe 4
	ENAA_P3	3	1	R/W	Enable auto acknowledgement data pipe 3
	ENAA_P2	2	1	R/W	Enable auto acknowledgement data pipe 2
	ENAA_P1	1	1	R/W	Enable auto acknowledgement data pipe 1
	ENAA_P0	0	1	R/W	Enable auto acknowledgement data pipe 0

APPENDICES

02	EN RXADDR				Enabled RX Addresses
	Reserved	7:6	00	R/W	Only '00' allowed
	ERX P5	5	0	R/W	Enable data pipe 5.
	ERX P4	4	0	R/W	Enable data pipe 4.
	ERX P3	3	0	R/W	Enable data pipe 3.
	ERX P2	2	0	R/W	Enable data pipe 2.
	ERX P1	1	1	R/W	Enable data pipe 1.
	ERX P0	0	1	R/W	Enable data pipe 0.
03	SETUP AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' - 5 bytes LSByte is used if address width is below 5 bytes
04	SETUP RETR				Setup of Automatic Retransmission
	ARD	7:4	0000	R/W	Auto Retransmit Delay '0000' - Wait 250µS '0001' - Wait 500µS '0010' - Wait 750µS '1111' - Wait 4000µS (Delay defined from end of transmission to start of next transmission) ^a
	ARC	3:0	0011	R/W	Auto Retransmit Count '0000' - Re-Transmit disabled '0001' - Up to 1 Re-Transmit on fail of AA '1111' - Up to 15 Re-Transmit on fail of AA
05	RF CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF CH	6:0	0000010	R/W	Sets the frequency channel nRF24L01 operates on

APPENDICES

06	RF SETUP				RF Setup Register
	Reserved	7:5	000	R/W	Only '000' allowed
	PLL LOCK	4	0	R/W	Force PLL lock signal. Only used in test
	RF DR	3	1	R/W	Air Data Rate '0' – 1Mbps '1' – 2Mbps
	RF PWR	2:1	11	R/W	Set RF output power in TX mode '00' – -18dBm '01' – -12dBm '10' – -6dBm '11' – 0dBm
	LNA HCURR	0	1	R/W	Setup LNA gain
07	STATUS				Status Register (In parallel to the SPI command word applied on the MOSI pin, the STATUS register is shifted serially out on the MISO pin)
	Reserved	7	0	R/W	Only '0' allowed
	RX DR	6	0	R/W	Data Ready RX FIFO interrupt. Asserted when new data arrives RX FIFO ^b . Write 1 to clear bit.
	TX DS	5	0	R/W	Data Sent TX FIFO interrupt. Asserted when packet transmitted on TX. If AUTO_ACK is activated, this bit is set high only when ACK is received. Write 1 to clear bit.
	MAX RT	4	0	R/W	Maximum number of TX retransmits interrupt Write 1 to clear bit. If MAX_RT is asserted it must be cleared to enable further communication.
	RX P NO	3:1	111	R	Data pipe number for the payload available for reading from RX_FIFO 000-101: Data Pipe Number 110: Not Used 111: RX FIFO Empty
	TX FULL	0	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
08	OBSERVE TX				Transmit observe register
	PLOS CNT	7:4	0	R	Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to RF_CH .

APPENDICES

	ARC CNT	3:0	0	R	Count retransmitted packets. The counter is reset when transmission of a new packet starts.
09	CD				
	Reserved	7:1	000000	R	
	CD	0	0	R	Carrier Detect.
0A	RX ADDR P0	39:0	0xE7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0B	RX ADDR P1	39:0	0xC2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0C	RX ADDR P2	7:0	0xC3	R/W	Receive address data pipe 2. Only LSB. MSBytes is equal to RX_ADDR_P1[39:8]
0D	RX ADDR P3	7:0	0xC4	R/W	Receive address data pipe 3. Only LSB. MSBytes is equal to RX_ADDR_P1[39:8]
0E	RX ADDR P4	7:0	0xC5	R/W	Receive address data pipe 4. Only LSB. MSBytes is equal to RX_ADDR_P1[39:8]
0F	RX ADDR P5	7:0	0xC6	R/W	Receive address data pipe 5. Only LSB. MSBytes is equal to RX_ADDR_P1[39:8]
10	TX ADDR	39:0	0xE7E7E7E7	R/W	Transmit address. Used for a PTX device only. (LSByte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device with Enhanced ShockBurst™ enabled.
11	RX PW P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P0	5:0	0	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
12	RX PW P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P1	5:0	0	R/W	Number of bytes in RX payload in data pipe 1 (1

APPENDICES

					to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
13	RX PW P2				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P2	5:0	0	R/W	Number of bytes in RX payload in data pipe 2 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
14	RX PW P3				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P3	5:0	0	R/W	Number of bytes in RX payload in data pipe 3 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
15	RX PW P4				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P4	5:0	0	R/W	Number of bytes in RX payload in data pipe 4 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
16	RX PW P5				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX PW P5	5:0	0	R/W	Number of bytes in RX payload in data pipe 5 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes

APPENDICES

17	FIFO STATUS				FIFO Status Register
	Reserved	7	0	R/W	Only '0' allowed
	TX REUSE	6	0	R	Reuse last transmitted data packet if set high. The packet is repeatedly retransmitted as long as CE is high. TX_REUSE is set by the SPI command REUSE_TX_PL, and is reset by the SPI commands W TX_PAYLOAD or FLUSH_TX
	TX FULL	5	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
	TX EMPTY	4	1	R	TX FIFO empty flag. 1: TX FIFO empty. 0: Data in TX FIFO.
	Reserved	3:2	00	R/W	Only '00' allowed
	RX FULL	1	0	R	RX FIFO full flag. 1: RX FIFO full. 0: Available locations in RX FIFO.
	RX EMPTY	0	1	R	RX FIFO empty flag. 1: RX FIFO empty. 0: Data in RX FIFO.
N/A	ACK_PLD ^c	255:0	X	W	Written by separate SPI command ACK packet payload to data pipe number PPP given in SPI command Used in RX mode only Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled first in first out.
N/A	TX_PLD	255:0	X	W	Written by separate SPI command TX data payload register 1 - 32 bytes. This register is implemented as a FIFO with three levels. Used in TX mode only
N/A	RX_PLD	255:0	X	R	Read by separate SPI command RX data payload register. 1 - 32 bytes. This register is implemented as a FIFO with three levels. All RX channels share the same FIFO
1C	DYNPD ^c				Enable dynamic payload length
	Reserved	7:6	0	R/W	Only '00' allowed
	DPL_P5	5	0	R/W	Enable dyn. payload length data pipe 5.

APPENDICES

					(Requires EN_DPL and ENAA_P5)
	DPL_P4	4	0	R/W	Enable dyn. payload length data pipe 4. (Requires EN_DPL and ENAA_P4)
	DPL_P3	3	0	R/W	Enable dyn. payload length data pipe 3. (Requires EN_DPL and ENAA_P3)
	DPL_P2	2	0	R/W	Enable dyn. payload length data pipe 2. (Requires EN_DPL and ENAA_P2)
	DPL_P1	1	0	R/W	Enable dyn. payload length data pipe 1. (Requires EN_DPL and ENAA_P1)
	DPL_P0	0	0	R/W	Enable dyn. payload length data pipe 0. (Requires EN_DPL and ENAA_P0)
1D	FEATURE ^c			R/W	Feature Register
	Reserved	7:3	0	R/W	Only '00000' allowed
	EN_DPL	2	0	R/W	Enables Dynamic Payload Length
	EN_ACK_PAY ^d	1	0	R/W	Enables Payload with ACK
	EN_DYN_ACK	0	0	R/W	Enables the W_TX_PAYLOAD_NOACK command

Reference: Nordic Semiconductor. (n.d.) *nRF24L01 Single Chip 2.4GHz Transceiver*. [e-book]

Nordic Semiconductor.