Intermediate Model-Based Design Training Program with VCU using MPC5744P

by

Ivan Lenskikh

Min Khant Soe

Sotheanith Vanny

A report submitted in partial fulfillment of the requirements for
the degree of Bachelor of Engineering in
Electrical and Electronics Engineering, Mechatronics Engineering and
Electrical Engineering

Project Advisor:

Asst. Prof. Dr. Narong Aphiratsakun

Examination Committee:

Dr. Jerapong Rojanarowan, Dr. Wisuwat Plodpradista,
Assoc. Prof. Dr. Jiradech Kongthon, Mr. Sunchanan Charanyananda,
Mr. Amulya Bhattarai, Mr. Ehsan Ali

Assumption University
Vincent Mary School of Engineering
Thailand

October 2020

Approved by Project Advisor:

Name: Asst. Prof. Dr. Narong Aphiratsakun

Signature:

Date: ______________________

Plagiarism verified by:

Name: Mr. Ehsan Ali

Signature:

Date: 02-Oct-2020

Acknowledgement

We are grateful and thankful to have Asst.Prof.Dr. Narong Aphiratsakun as our guidance and advising for this project. We also would like to thank to Arjarn Warongkidh Ganchanasopa that instructs us with safety and protection function with high voltage application testing. We sincerely thank to Mine Mobility Research that trust and sponsor for this project to be able to work.

Abstract

Due to the numerous environmental problems, one of them being air pollution, the advancement in electric cars becomes one of the highest priorities in many countries. In an electric car, all equipment is controlled by a Vehicle Control Unit (VCU). Thus, understanding how to program a VCU is an important task in the electric car industry. We plan to create a training course for students at Vincent Mary School of Engineering at the Assumption University that can be added to the current curriculum.

This project is an intermediate level training course with a two-day duration which can be conducted in a weekend. In this training course, NXP (Next Experience) devkit-MPC5744P will be used as a development microcontroller board which is the board that focuses on the automotive, industrial safety function, motor control, hybrid or electric vehicle power inverter, powertrain and Vehicle Control Unit (VCU). The practical examples as well as the experiments with industrial equipment will be demonstrated on NXP devkit-MPC5744P. This project will assist and prepare students for working with electric car companies as an intern as well as preparing students for future school projects.

Table of Contents

1. Chapter 1

1.1. Introduction

With the high risk of environmental problems due to high carbon emission, Electric Vehicles (EV) are the future of the automobile industry. To build an electric vehicle, the most crucial part, the vehicle control unit, cannot be missed. It can be described as the brain of electric vehicles. VCU controls most of the functions of electrical vehicles. Since the devkit-MPC5744P microcontroller board and VCU have the same processor, they both can be used for the same purpose. However, the microcontroller can be used only for experiments and developments; it cannot be directly used to implement and control the components of EV like VCU does. As engineering students, we need to prepare for future careers in the automobile industry that would require many engineers to build and design systems using model-based design and even low-cost microcontrollers.

1.2. Project Overview

This project's objective is to create an intermediate level design training course that can be conducted for two days on a weekend. In this training course, NXP (Next Experience) devkit-MPC5744P will be used as a development microcontroller board which is the board that focuses on automation, industrial safety, motor control, hybrid, or electric vehicle power inverter, powertrain and Vehicle Control Unit. The practical examples as well as the experiments with industrial equipment will be demonstrated on this NXP devkit-MPC5744P.

The learning practicum will be started with the basic SIMULINK environment of writing and testing the devkit-MPC5744P. Moreover, VCU will be used for motor speed control and to control the output voltage of the DC-DC converter.

1.3. Background Study

Mine Mobility Research (MMR) is the research company that focuses on electric vehicles under Energy Absolute. Energy Absolute (EA) is one of the main energy companies that design and generate electrical power in Thailand. Furthermore, EA is designing and manufacturing its own electric vehicles. MMR designs and constructs the electric vehicle based on Model-Based Design using VCU which has MPC5744P microprocessor. MMR is the future eco-friendly technology company that is planning to develop all modes of transportation using electricity including mass-transit transport. It requires engineers who know the model-based design to create complex electric vehicle systems.

One approach in model-based design with SIMULINK environment was conducted in Assumption University by Asst.Prof.Dr. Narong Aphiratsakun with Arduino microcontrollers. Arduino microcontroller is not suitable to use in the automobile industry which requires a more stable microcontroller. Arduino microcontrollers are mostly used for the basic experiments and for gaining practical knowledge of model-based design before going to the industrial microcontroller such as NXP devkit-MPC5744P.

Figure 1.3.1 Group Photo During Basic Model-Based Design Training

Figure 1.3.2 Recorded Photo during the Basic Model-Based Design Training

1.4. Project Layout

1.4.1. Basic Model-Based Design

Engineers and scientists use many methodologies to design and construct systems for the automobile. Among those design methods, Model-Based Design (MBD) is one of the tools that is being used in the automotive industry. MBD is mainly used in SIMULINK MATLAB software which has the plant model, analysis and can convert into a mathematical model by code generation. MBD is suitable for all controllers that are supported by the MATLAB package.

1.4.2. Model-Based Design of Vehicle Control Unit

Companies that produce hybrid or electric vehicles (EV) use Model-Based Design as an effective tool to create different vehicle control systems. The basic electric component of an electric vehicle is the Electronic Control Unit (ECU). An ECU is a special type of embedded system that is dedicated to controlling a specific aspect of a vehicle. The Vehicle Control Unit (VCU) is a type of ECU. VCU is the main electronic component of an electric vehicle which is capable of overseeing other ECU’s work and to control some of the EV functions as well. First, we would need to learn the architecture of VCU and its SIMULINK capabilities. The VCU is much more complex than NXP devkit-MPC5744P alone. Then after getting familiar with the hardware, we will conduct a range of experiments on VCU using SIMULINK to advance even more our knowledge about this technology.

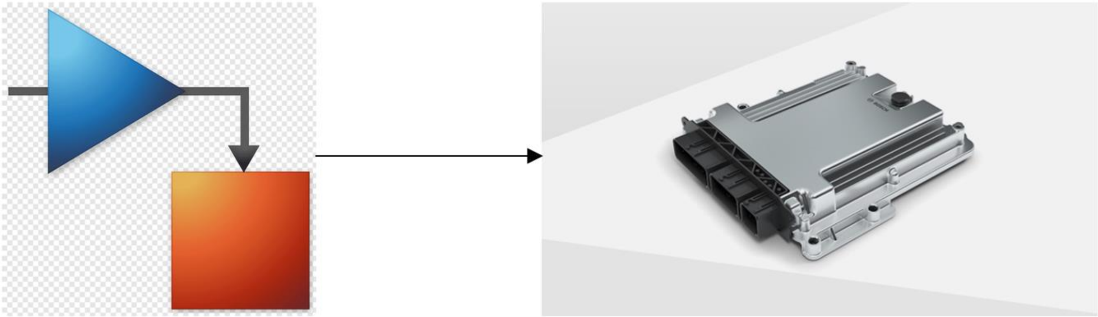


Figure 1.4.1 Model-Based Design of Vehicle Control

1.4.3. Model-Based Design for Electric Vehicle

In a factory-grade electric vehicle, functions of the Vehicle Control Unit are not fixed. The VCU is a very powerful equipment that is capable of not only supervising other systems but to duplicate some of their functions as well. At this stage, we will implement different functions of the VCU using factory grade hardware. We will create the software using Model-Based Design to execute some of the VCU's tasks. By doing this we will build real-life systems for an electric vehicle.

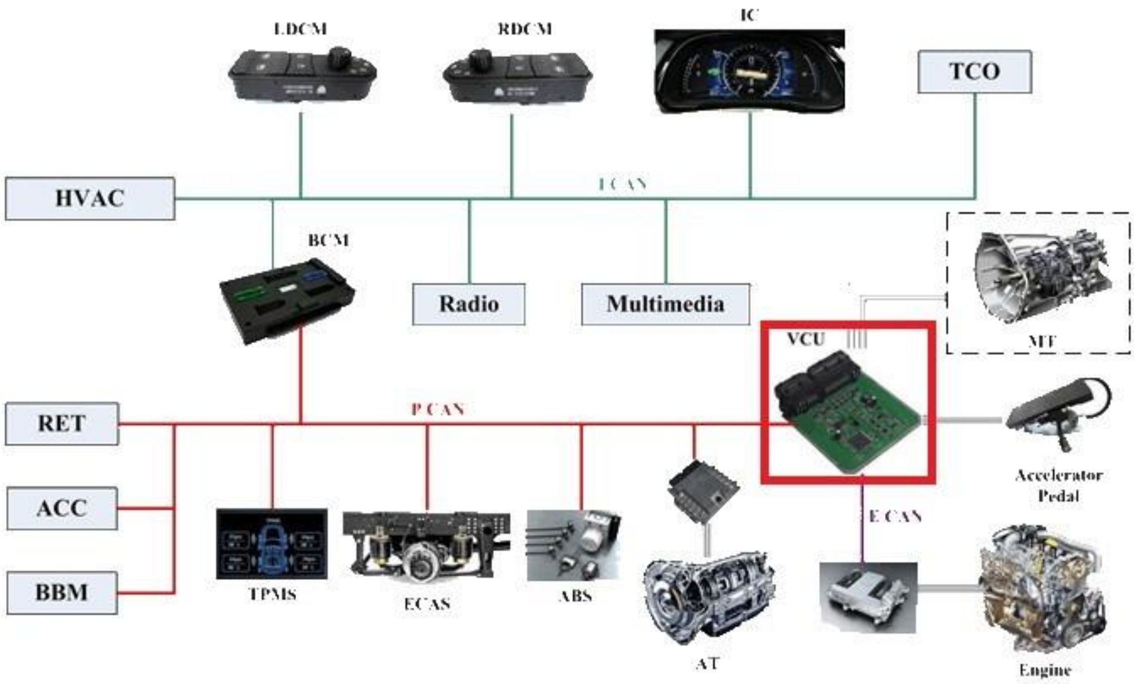


Figure 1.4.2 Schematic for VCU position in an Electric Vehicle control system

1.5. Objective

The main objective of this project is to develop a training course using a model-based design system with an industry-grade VCU and devkit-MPC5744P board. The course aims to provide the knowledge of MATLAB SIMULINK (graphical coding) relating to the devkit-MPC5744P board and the VCU. The course hopes to assist VME students to work with automobile companies for their senior project since they will learn how to use a VCU in EV; it is also helpful for applying to jobs in the automobile industry.

2. Chapter 2

2.1. Proposed Training Session Topics

Seven topics that are arranged from basic to intermediate level will be constructed for the model-based design using NXP devkit-MPC5744P and vehicle control unit.

No.	Topics	Subtopics
1.	Flash the bootloader of MPC5744P	- Download the program into NXP and VCU
2.	Digital and Analog Input/Output Control	- Digital input/output - Multiple digital inputs/outputs - 7-segment display - Potentiometer controls the intensity of Light Emitting Diode (LED) with Pulse Width Modulation (PWM)
3.	Timer and Alarm	- Introduction to timer and alarm - Application of Timer and Alarm: Safety System of Motor Temperature
4.	DC Stepper Motor Control	- Direction and Speed control of motor using ultrasonic sensor
5.	Controller Area Network (CAN) Communications	- CAN communication using GY8507 - CAN communication between microcontrollers
6.	Brushed DC Motor Control	- Using VCU to control speed of Single-phase DC motor
7.	Electric Vehicle DC – DC Converter Control	- Control the output voltage with CAN protocol

Table 2.1.1 List of the Topics

2.2. Controllers

2.2.1. Devkit - MPC5744P Board

The NXP Devkit - MPC5744P is a development platform for MPC5744P microcontrollers, developed by NXP Semiconductor. This board is designed for industrial safety functions, automotive and motor control applications. It is the low-cost microcontroller that is being used in electric vehicle systems and industrial factories. The microcontroller is built with a pinout that can have many functions for the same pins. This board has a large number of applications that are suitable for usage, especially in EV and powertrain.

Features

- 2 x 200 MHz Power Architecture 200Z4 Dual issue cores
- AEC-Q100 Grade 1
- Ambient temperature: -40 to +125 °C
- Support motor control shield (DEVKIT-MOTORGD)
- Integrated Open-Standard Serial and Debug Adapter (OpenSDA) with support for use of industrial standard debugging interfaces
- JTAG connector
- Ease of access to the MCU I/O header pins for prototyping
- FlexRay, CAN, LIN, UART/SCI and SPI
- On board potentiometer
- On board RGB LED
- Two on board push-button switches
- Power supply: micro-B USB and 12V External power supply

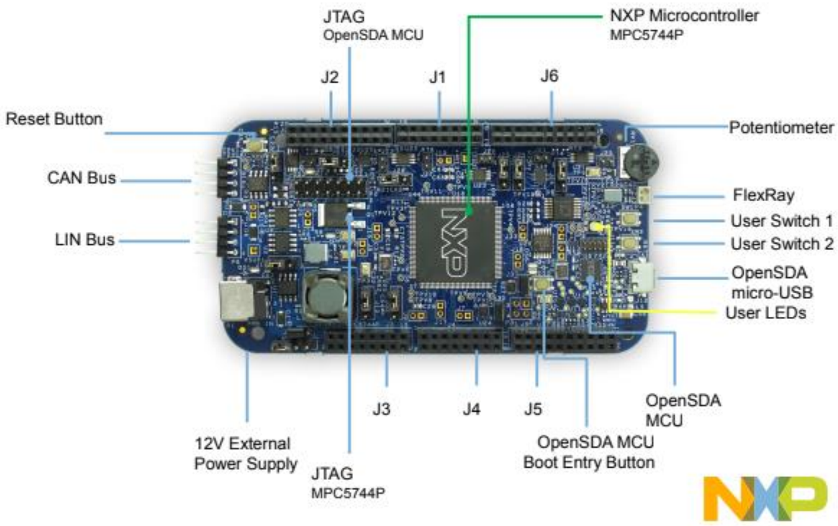


Figure 2.2.1 NXP Devkit-MPC5744P

Name	Pin Numbers
Digital I/O	PA 0-15, PB 0-15, PC 0-2, 4-7, 10-15 PD 0-2, 4-7, PI 0-15 PG 2-11, PE 0, 2, 4-7, 9-15 PF 0, 3, 4-15 PG 2-11 PH 4-15 PJ 0-9
Analog I/O	PB 7 -13
CAN	PA 14, 15 PB 0,1

Table 2.2.1 Pin Data Sheet for Devkit-MPC5744P

2.2.2. Vehicle Control Unit (VCU)

Vehicle Control Unit is the main controller for an electric or hybrid vehicle. VCU plays a critical and supervisory role in the whole vehicle control network. Apart from supervising other Electric Control Units, VCU usually functions as a receiver for the sensors and a driver input signals (speed and brake pedals), manages the system energy, communicates with the Engine Control Unit, coordinates between different ECUs, check for system errors and determines the overall vehicle status. For this project, we are using a VCU based on the NXP MPC5744. After analyzing the datasheet, we can summarize its main features:

Figure 2.2.2 Vehicle Control Unit (VCU) Board

- Microcontroller unit (MCU) is NXP MPC5744
- Inputs: 15 analog, 21 digital, 4 frequency and 3 wake-up
- Outputs: 10 high-side drivers (2 PWM) and 18 low-side drivers (4 PWM)
- Operating voltage is 9-32V
- Communication: 3 CAN 2.0B
- Sensor 5V supply: 5 channels
- Operating temperature: -40°**C** to +85°**C**
- ISO16750 compliant
- Simulink Model Based Design

Description	PIN Number	Specification
Digital Inputs	26, 43 ,45, 63, 65, 67, 68	Active-high: ≥8.5V. Input voltage range: 0-24V
Digital Inputs	21, 25, 31, 38, 42, 52 ,53	Active-low: ≤5V. Input voltage range: 0-24V
Analog Inputs	16, 17, 22, 23 ,24, 35 ,71, 72	A/D resolution: 12bit. Input voltage range: 0-36V Dividing resistor: 3.48K
Analog Inputs	18, 36	A/D resolution: 12bit. Input voltage range: 0-5V Pull-up resistor: 10K
Analog Inputs	20, 37, 61, 62, 79	A/D Resolution: 12bit. Input voltage range: 0-5V
DC 12/24 V Power	1, 3, 116, 119	Voltage range: 9-32V
Ground	2, 4, 5, 120, 121	VCU ground
Frequency Inputs	8, 47, 64, 66	Input frequency range: 1Hz – 5kHz (can be configured as digital inputs)
High-side drivers	85, 86	1A rated, peak current 3A (can be configured as PWM outputs 20Hz – 1KHz)
Low-side drivers	90, 91, 92, 93	0.5A rated, peak current 1A (can be configured as PWM outputs 20Hz – 1KHz)
CAN communication	54, 55, 56, 57, 58, 73, 76, 77	Built-in 120 Ω terminal resistor (pins 54, 55, 56, 73)
5V Sensor Supply	6, 41, 49, 51, 69	Maximum current: 50mA. Voltage supply: 5V±2%
Sensor Ground	48, 50, 60, 70, 74, 75	VCU sensor ground

Table 2.2.2 Pin Data Sheet for Devkit-MPC5744P

2.3. Software Used

2.3.1. MATLAB SIMULINK

Figure 2.3.1 Main Software

MATLAB is an advanced and productive software (created by MathWorks) that has many functions and applications. It will be the main software in this project. SIMULINK is the feature used in MATLAB to transform the block modeling programming into a mathematical model. SIMULINK has its libraries that are the tools of model-based design.

2.3.2. NXP Model Based Design Toolbox for MPC57xxP

Model-Based Design Toolbox is a MATLAB SIMULINK toolbox created by NXP company to write the programs and work with NXP microcontroller boards such as MPC5744P. It can be used to execute many complex tasks such as CAN, UART, motor control, and sensor-based applications on NXP MCUs.

Features:

- Generate code for stand-alone application with direct download to target support
- Optimized motor control library blocks for Park/Clarke transforms, digital filters, and general functions
- I/O blocks for Analog, Digital, CAN, Serial Peripheral Interface (SPI), programmable interval timer (PIT), Sine Wave Generation, Event timer and PWM.
- On-target profiling of functions and tasks
- Bootloader utility for programming application in flash
- Seamless integration with embedded coder for software-in-the-loop (SIL) and processor-in-the-loop (PIL) testing

2.3.3. S32 Design Studio for Power Architecture

Figure 2.3.2 Flash Tool

It is a high-quality software tool (created by NXP) that is used to flash the bootloader for Devkit - MPC5744P and VCU. S32 design studio allows the download of Simulink file into the board faster than serial port.

2.3.4. CAN Tool Analyzer GY8507

CAN tool is one of the software that use to interface CAN bus through serial port to the host computer. CAN tool software provide by Glinker specifically for their own CAN

USB adapter. The software can be used for CAN bus product develop, monitoring and data analysis.

2.4. Theory

2.4.1. Pulse Width Modulation (PWM)

Pulse-width modulation (PMW) is the concept used in analog control applications. PWM is a way of delivering energy through a series of pulses rather than a continuously varying analog signal which will in effect reduce the average power delivered by an electrical signal. By increasing or decreasing pulses width, the controller regulates energy flow to the load. The load's inductance acts then like a filter, storing energy during the "on" cycle while releasing it at a rate corresponding to the input or reference signal. The switching rate (frequency) varies greatly depending on load and application.

PWM control is widely used in power electronic applications such as converter duty cycle control, communication systems and digital controls. In this project, we use it to control the brightness of the LED and the speed of the servo motor. It allows us to control how long the signal will be high or low and we can calculate by using the duty cycle. The term duty cycle describes the proportion of "on" time to the regular interval (period). The duty cycle is expressed in percent, 100% being fully on. For example, when a digital signal is on half of the time and off the other half of the time, the duty cycle of 50%.

We can produce a PMW signal using a comparator operational amplifier. By supplying a sawtooth carrier as one of the inputs and sinusoidal signal as the other. The output will stay high as long as the command is greater than the carrier. The main advantage of using PWM is its efficiency, the power loss in the switching devices is very low. Also, PWM amplifiers run cooler than standard linear power amplifiers requiring less heat sink. But the main disadvantage is that high-current switching generates electromagnetic noise as well as voltage spikes. This calls for special measures like filtering, shielding, etc.

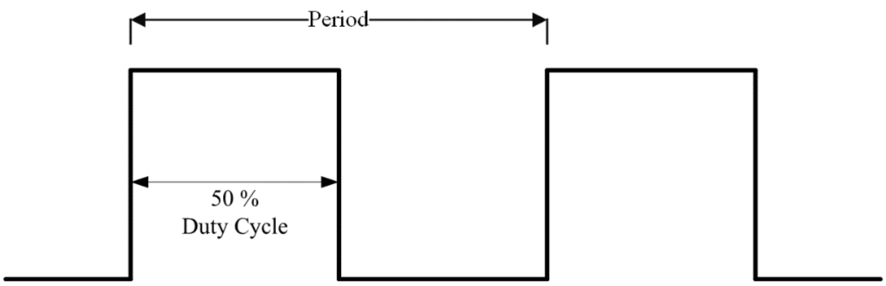


Figure 2.4.1 Pulse Width Modulation Waveform

The Devkit-MPC5744P board provides 3.3 V and the analog output signal (0-255), 50 % duty cycle of 3.3 V will be 1.65 V and the analog output signal will be 127.5. By calculating the duty cycle, we can find the value to control the brightness of the LED and speed of the servo motor.

2.4.2. CAN Communication

Controller Area Network (CAN Bus) is a vehicle standard. We can use it to communicate between electronic control units and other devices even without a host computer. By using the CAN Bus, we reduce the number of wires needed to connect for each device. It can be used for a centralized control system using one microcontroller to control several devices. We can also use CAN Bus to communicate (data sending and receiving) between two microcontrollers.

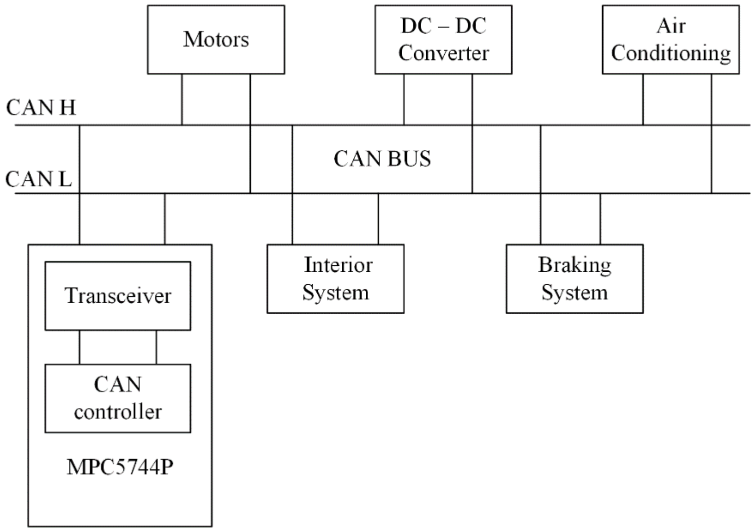


Figure 2.4.2 CAN Bus connection diagram in Automotive

There are 4 types of CAN frames.

Type of frames	Description
Data frame:	It is for actual data transmission
Remote frame:	It is consequently followed by a data frame and contains the requested data.
Error frame:	Any sender or receiver will report error conditions during data frame or remote frame transmission.
Overload frame:	It will report node overload and provide delay between data or remote transmission.

Table 2.4.1 Four types of CAN frames

The data frame is for actual data transmission. It has two formats which are base frame (with 11 identifier bits) and extended frame (with 29 identifier bits).

Name	SOF	Identifier	RTR	Control	Data	CRC	ACK	EOF
Number of bits	1	11 or 29	1	6	0 - 64	16	2	7
		Arbitration Field						

Table 2.4.2 CAN Message Frame

Name	Description		
Start of frame (SOF)	CAN message begins from this stage.		
Identifier	It indicates the priority. Lower value is higher priority.		
Remote Transmission Request (RTR)	RTR must be “0” for the data frame. RTR must be “1” for remote request frames.		
Control	Identifier extension bit (IDE)	-	For 11 bits identifier: IDE must be “0” - For 29 bits identifier: IDE must be “1”.
	Reversed bit (R0)	-	R0 must be “0”.
	Data length code (DLC)	-	It contains the number of bytes of data being transmitted.
Data field	Data field contains actual data which is to be transmitted.		
Cyclic Redundancy Check (CRC)	CRC is error detecting code and ensures data integrity.		
ACK	ACK slot	-	Transmitter sends “1”. - Any receivers can assert “0”.
	ACK delimiter	-	It must be “1”.
End of Frame (EOF)	EOF marks the end of the frame and must be “1”.		

Table 2.4.3 CAN Message Frame Explanation

In CAN Bus Arbitration, dominant bit will override recessive bit where bit “0” = dominant bus level and bit “1” = recessive bus level.

Node			BUS
X	Y	Z	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table 2.4.4 Explanation of CAN Bus Arbitration

If two or more nodes are “0”, the node which transmits the lowest message ID will win the arbitration, so the message will be transmitted. Nodes which lost arbitration will start a new arbitration when the CAN bus is free for access again.

2.4.3. Feedback Control of a DC motor with a PID controller

To control the speed of a DC motor with a feedback system, we need to find a transfer function of that DC motor.

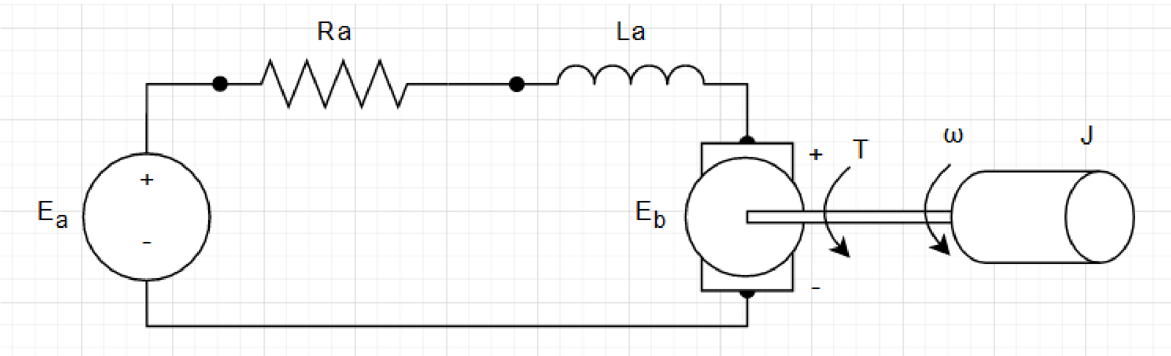


Figure 2.4.3 Equivalent circuit of an armature-controlled DC motor

For this circuit, field current must be constant and armature current can be controlled through armature voltage, E_a.

e_a = Armature voltage of DC motor
e_b = Back electromotive force (emf) of DC motor
i_a = Armature current
R_a = Resistance of armature winding

L_a = Inductance of armature winding
T_d = Torque developed by motor
T_L = Load torque
K_E = Back emf constant
K_T = Developed torque constant
ω_m = Speed of DC motor
J_m = The moment of inertia of DC motor referring to the motor shaft
B = Viscous-friction coefficient of DC motor referring to the motor shaft

Back emf of DC motor depends on the rate of rotation of that motor.

$$e_b(t) = K_E\omega_m(t) \quad \rightarrow \quad E_b(s) = K_E\omega_m(s) \qquad \text{eq (1)}$$

Torque developed by motor is directly proportional to the armature current.

$$T_d(t) = K_t\, i_a(t) \quad \rightarrow \quad T\_d\,(s) = K_T\, I_a(s) \qquad \text{eq (2)}$$

The equation of armature circuit of a DC motor:

$$e_a(t) = R_a\, i_a(t) + L_a \frac{di_a}{dt} + e_b(t)$$

The equation of armature circuit of a DC motor in Laplace transform

$$E_a(s) = R_a I_a(s) + L_a\, s\, I_a(s) + E_b(s)$$

$$I_a(s) = \frac{E_a(s) - E_b(s)}{(R_a + L_a\, s)} \qquad \text{eq (3)}$$

The equation of torque equilibrium

$$T_d(t) = J_m \frac{d\omega_m(t)}{dt} + B_m\, \omega_m(t) + T_L(t)$$

The equation of torque equilibrium in Laplace transform

$$T_d(s) = J_m\, s\, \omega_m(s) + B_m\, \omega_m(s) + T_L(s)$$

$$\omega_m(s) = \frac{T_d(s) - T_L(s)}{(J_m\, s + B_m)}$$

For no-load, $T_L = 0$,

$$\omega_m(s) = \frac{T_d(s)}{(J_m\, s + B_m)} \qquad \text{eq (4)}$$

Substitute eq (2) into eq (4)

$$\omega_m(s) = \frac{K_T\, I_a(s)}{(J_m\, s + B_m)} \qquad \text{eq (5)}$$

Substitute eq (3) into eq (5)

$$\omega_m(s) = \frac{K_T \left(\frac{E_a(s) - E_b(s)}{(R_a + L_a\, s)} \right)}{(J_m\, s + B_m)} \qquad \text{eq (6)}$$

$$\frac{\omega_m(s)}{E_A(s)} = \frac{K_T}{(R_a + L_a\, s)(J_m + B_m) + K_T K_E}$$

$$\frac{\omega_m(s)}{E_A(s)} = \frac{K_T}{R_a\, J_m s\; + R_a B_m + L_a\, J_m\, s^2 + B_m L_a\, s + K_T K_E}$$

Transfer function of speed of a DC motor is obtained.

$$\frac{\omega_m(s)}{E_A(s)} = \frac{K_T}{L_a\, J_m\, s^2 + (R_a\, J_m + B_m L_a\,)\, s\; + (R_a B_m + K_T K_E)}$$

To achieve a steady state error of zero for a step input and to control both maximum overshoot and rise time, PID controller is needed to be used. The integral controller eliminates the steady state error and determines the settling time. Proportional and derivative controller can control the maximum overshoot and rise time.

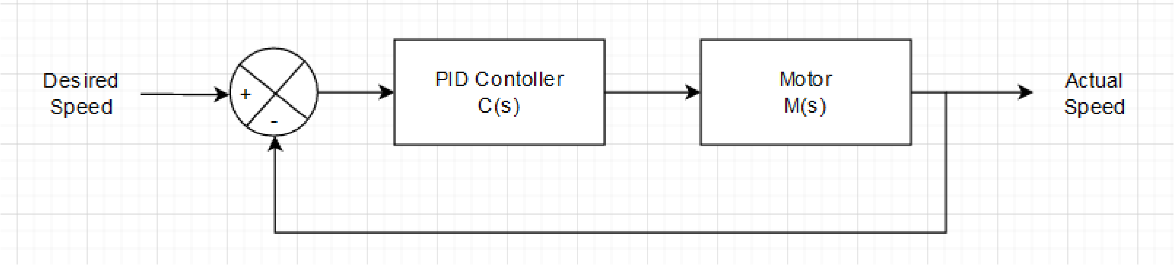


Figure 2.4.4 Closed Loop speed control of a DC motor with PID controller

To avoid the complication in finding transfer function of the closed-loop speed control of DC motor with PID controller, we can simply the transfer function of DC motor in this way.

Let $A = L\, J_m$
Let $B = R_a J_m + B_m L$
Let $C = R_a B_m + K_T K_E$

Transfer function of speed of a DC motor:

$$M(s) = \frac{\omega_m(s)}{E_A(s)} = \frac{K_T}{As^2 + Bs + C}$$

Transfer function of a PID controller:

$$C(s) = K_p + \frac{K_i}{s} + K_d\, s$$

Transfer function of opened loop control of a DC motor speed control system:

$$\mathrm{G(s) = C(s) * M(s)} = \frac{\left(\mathrm{K_p} + \frac{\mathrm{K_i}}{\mathrm{s}} + \mathrm{K_d\ s}\right)\ K_T}{As^2 + Bs + C}$$

Transfer function of unity feedback (closed loop) control of a DC motor speed control system

$$T(s) = \frac{G(s)}{1 + G(s)} = \frac{K_T\ (K_d s^2 + K_p s + K_i)}{As^3 + (B + K_d K_T)s^2 + (C + K_P K_T)s + K_i K_T}$$

To estimate the values of K_p, K_i and K_d, we can set Ki = 0, and get estimated values of K_p and K_d. When we set K_i, transfer function of the system will become:

$$T(s) = \frac{K_T\ (K_d s + K_p)}{As^2 + (B + K_d K_T)s + K_P K_T C} \qquad \text{eq (7)}$$

ξ = Damping ratio
ω_n = Natural frequency
M_p = Maximum overshoot
T_r = Rise time

$$M_p = e^{-\pi\left(\frac{\xi}{\sqrt{1-\xi^s}}\right)} \qquad \text{eq (8)}$$

$$T_r = \frac{\pi - tan^{-1}(\frac{\sqrt{1-\xi^2}}{\xi})}{\omega_n\sqrt{1-\xi^2}} \qquad \text{eq (9)}$$

To guess the estimated values of K_p and K_d, we need to decide what is our rise time and maximum overshoot of 2nd order system, then we can find ω_n from eq (8) and ξ from eq (9).

$$2^{\mathrm{nd}}\ \text{order transfer function} = \frac{\omega_n^s}{s^2 + 2\xi\omega_n s + \omega_n^s} \qquad \text{eq (10)}$$

Then we can equate the denominator of eq (7) and eq (10), we will get the equations for K_p and K_d.

$$As^2 + (B + K_d K_T)s + K_P K_T C = s^2 + 2\xi\omega_n s + \omega_n^s$$

$$(B + K_d K_T) = 2\xi\omega_n \quad \rightarrow \quad K_d = \frac{2\xi\omega_n - B}{K_T}$$

$$K_P K_T C = \omega_n^s \quad \rightarrow \quad K_p = \frac{\omega_n^s}{K_T C}$$

When we get estimated values of K_p and K_d, we can tune PID control by varying K_p, K_i and K_d values in MATLAB.

Closed-loop response	Rise time	Overshoot	Settling time	Steady-state error	Stability
Increasing K_p	Decrease	Increase	Small increase	Decrease	Degrade
Increasing K_i	Small decrease	Increase	Increase	Large decrease	Degrade
Increasing K_d	Small decrease	Decrease	Decrease	Minor change	Improve

Table 2.4.5 The effect of PID tuning on closed loop system response

2.4.4. Switching – Mode Regulators

Switching – Mode Regulators are the dc converters that are used to regulate the output voltage with pulse width modulation (PWM) controlling of the switching frequency. The duty cycle controls the “on” time of the switch. There are four types of topologies in switching – mode regulators. Buck converter is the dc – dc step down converter which has the output voltage less than the input voltage and output current is higher than the input current. The buck converter is one of the topologies in switching – mode regulators. This converter is widely used in many dc applications such as battery system of electric vehicles and in many other appliances.

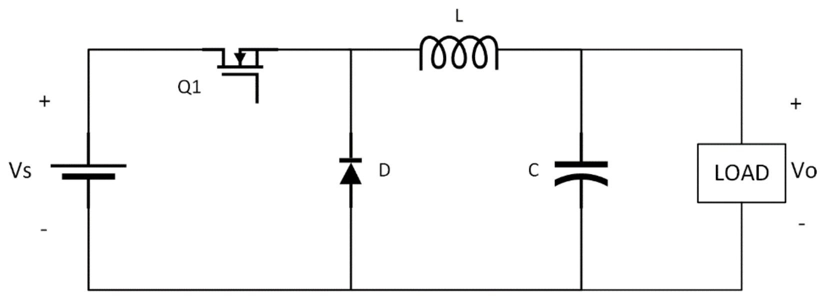
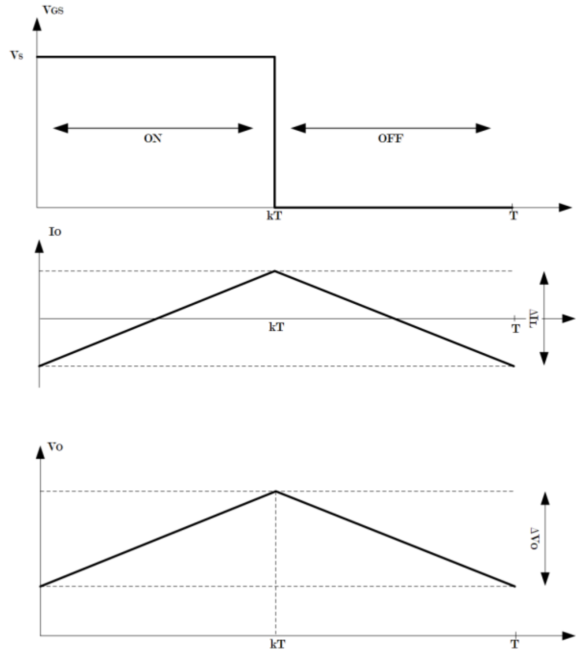


Figure 2.4.5 Topology of Buck Converter

A buck converter circuit consists of MOSFET (Q1), diode, inductance, capacitance and load connection are parallel to the capacitance. Two modes of buck converter are occurred when MOSFET is on/off. The output of the buck converter depends on the duty cycle (turn – on time) of the MOSFET. When the duty cycle increases, the output voltage decreases. In practical, the output voltage and current include ripple but those output are continuous. The average output voltage as:

$$V_a = V_s \frac{ON}{T} = kV_s$$



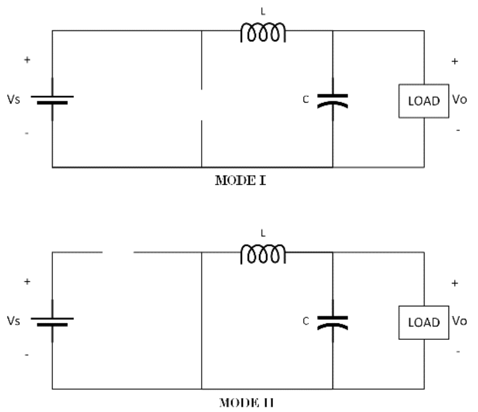


Figure 2.4.6 Two Modes of Buck Converter and Waveform

3. Chapter 3

3.1. Flash the Bootloader of MPC5744P

MPC5744P microcontroller series have several techniques for program to flash into its bootloader. The most common way to flash the program into the bootloader is using serial communication port. Serial communication port allows the microcontroller to communicate to the host computer and be able to transfer the data from host to destination. Yet, NXP semiconductor designed a platform which is specifically used for automotive and ultra-reliable microcontroller called S32 Design Studio Integrated Design Platform that released with vary type of platform. The software has no limits for developer to codes, compile and debugging. The microcontroller of MPC5744P used the platform called S32 Design Studio for Power Architecture. With this software, it is more efficient for developer and having less time of the development. Also, NXP created the SIMULINK model – based design toolbox that has the target for all of their development boards.

The target block is the set-up configuration for the model that can let the SIMULINK block model download into the NXP devkit-MPC5744P.

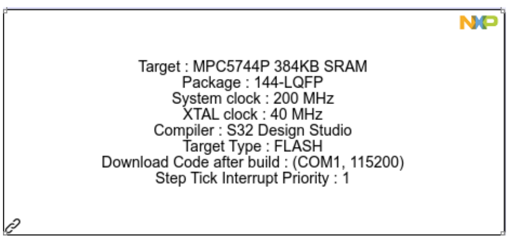


Figure 3.1.1 Target block of MPC5744P in Simulink

In this block, define the setup specification for MPC5744P microcontroller board. Target block requires for building the programs into the board.

3.1.1. Serial Communication Port

The serial communication port can be connected between host computer and MPC5744P target by using micro – USB and JTAG. With micro – USB, the program can flash directly from SIMULINK by using the target block from the NXP model – based designed toolbox.

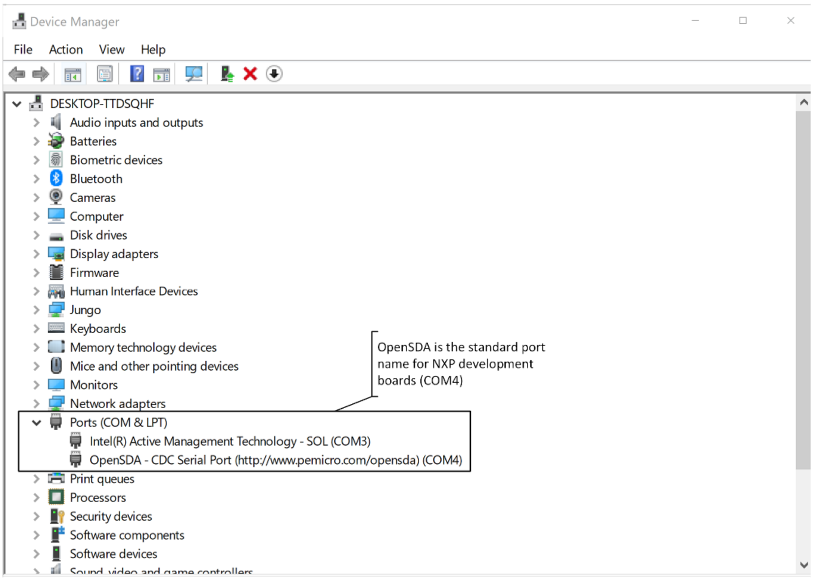


Figure 3.1.2 Device Manager Ports

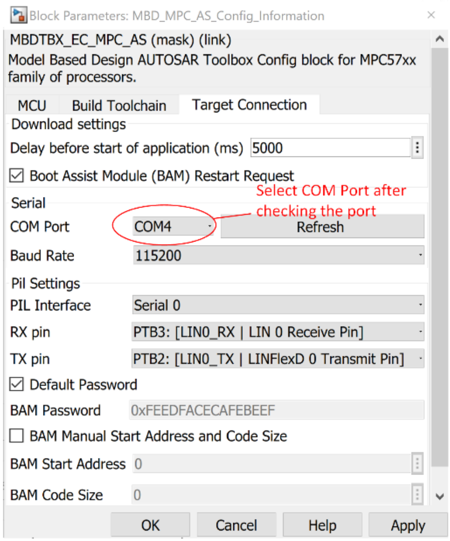


Figure 3.1.3 Define COM Port in Target Block

Device manager allows the users see the ports of their serial communication device connect to. NXP development boards come with OpenSDA port that could be recognized by the user instantaneously. Double click on the target block, block parameters will let the user to select the COM Port. Select the port of the communication is very crucial, without the port it could cause the building of Simulink failure.

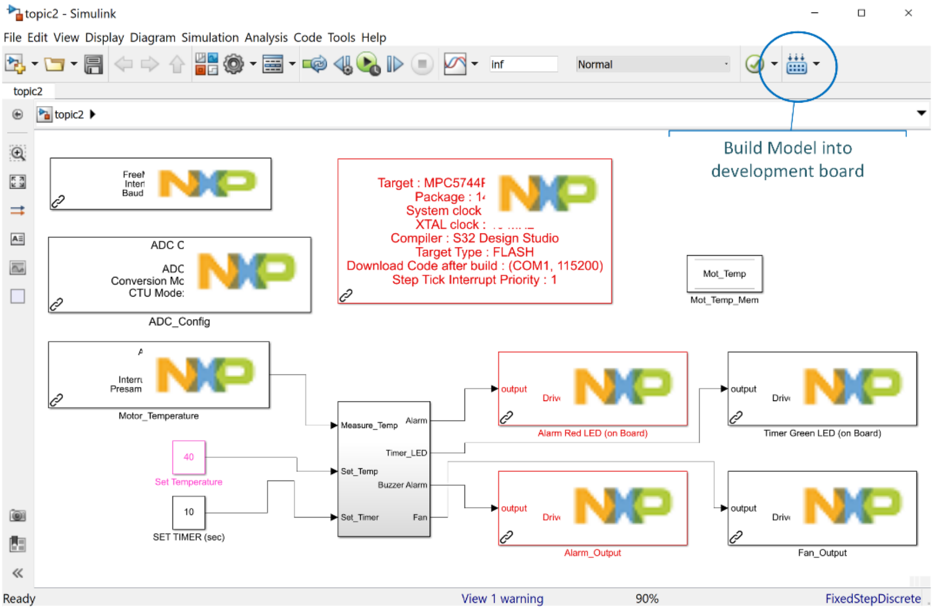


Figure 3.1.4 Build Model into Development Board

After the graphical program finished on the Simulink, the program needs to transfer these codes into the development board. To do this, click on the icon as shown above in *Figure 3.4* or using the shortcut on the keyboard *Ctrl+B*.

The Simulink message box will pop-up to let the user need to know the reset of the board before entering the new program.

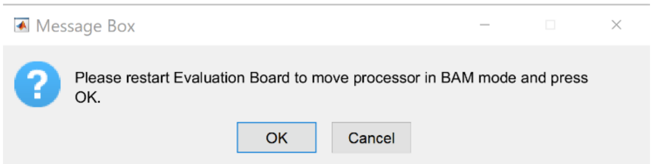


Figure 3.1.5 Reset Message

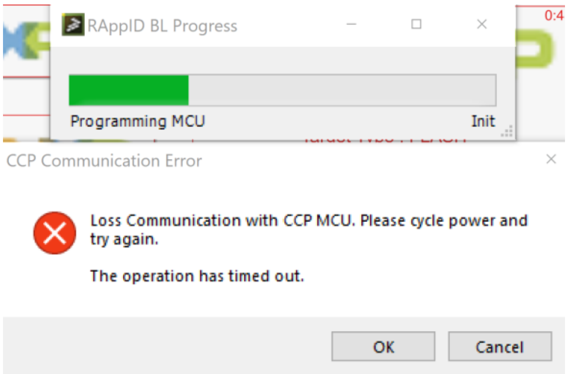


Figure 3.1.6 Error Message During Flashing

The error message occurred due to loss communication between the development board and host computer. The cause of error can be happened with the COM Port selection in the target block. Re-check the port of the development board.

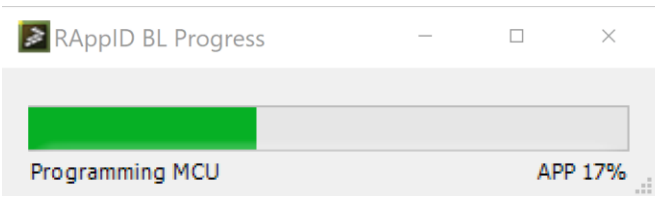


Figure 3.1.7 Successfully Build Model

The Simulink model successfully upload into the board when the message occurred as shown in *figure 3.7*.

3.1.2. JTAG - S32 Design Studio for Power Architecture

JTAG stands for joint test action group which is used to communicate from the host computer directly to the processor on the board. With the help of S32 Design Studio, flashing the MPC5744P with through JTAG without having the error occur and reset the chip before download the new program in.

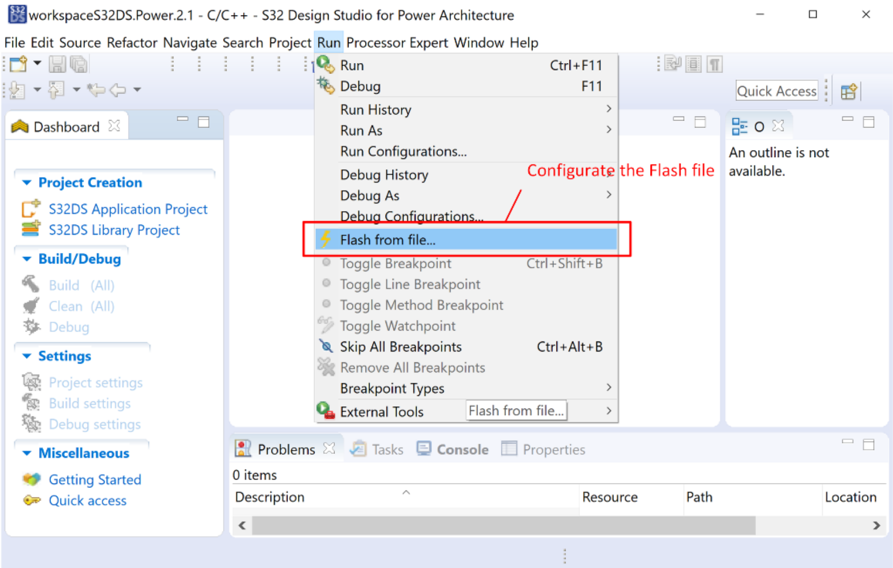


Figure 3.1.8 Configurate Flashing File

To insert the file for flashing the MPC5744P, the new configuration of the flashing file is needed to create.

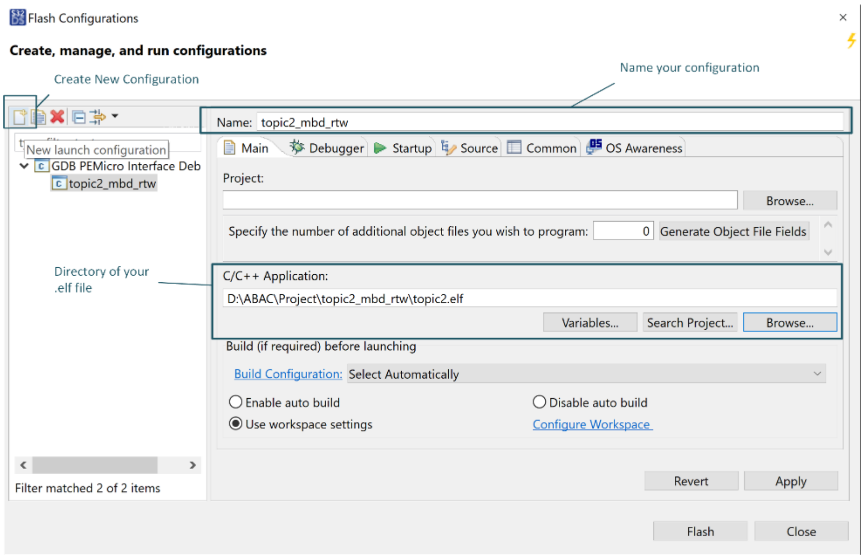


Figure 3.1.9 New Launch Configuration

The configuration window let the user create the flash tool, define the name of their project and define the directory of the .elf file. After building the Simulink model, Simulink will store the .elf file in the build folder of the model. Define target to flash is significant, setting up the name of the device, interface for communicating with the board and port of connection. Port of connection will detect automatically whenever the interface cable is connected. Choose the device name as MPC5744P with Z4 core, this configuration depends on the board that user is using.

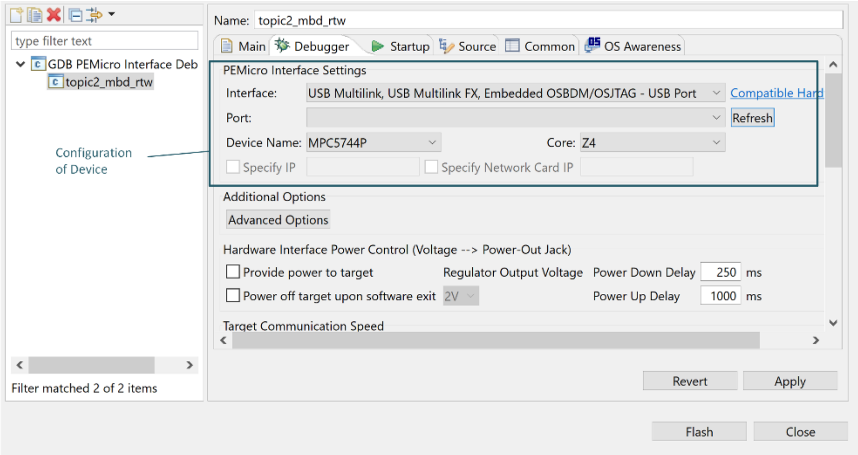


Figure 3.1.10 Target Configuration

After all of the configuration finished, click on the flash button that would transfer all the data into the processor directly.

3.2. Digital and Analog Input/Output Control of Devkit – MPC5744P and VCU

3.2.1. Digital Input/Output

This simple digital input and output program is designed for testing digital input and output pins and for gaining an understanding of a basic concept of SIMULINK block from the MBD toolbox for MPC5744P board. In this program, we try to test the active low digital input. When the switch is pressed, an LED will be OFF. Otherwise, LED will be ON. To make this program, we use NOT logic operation. Using these concepts, we can develop a program to turn ON and OFF the devices such as a motor.

SW	LED
0	1
1	0

Table 3.2.1 Truth table for NOT operation

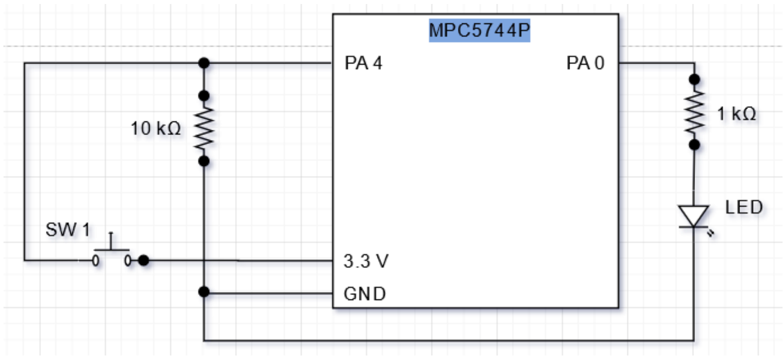


Figure 3.2.1 Circuit diagram of Digital Input and Output Program

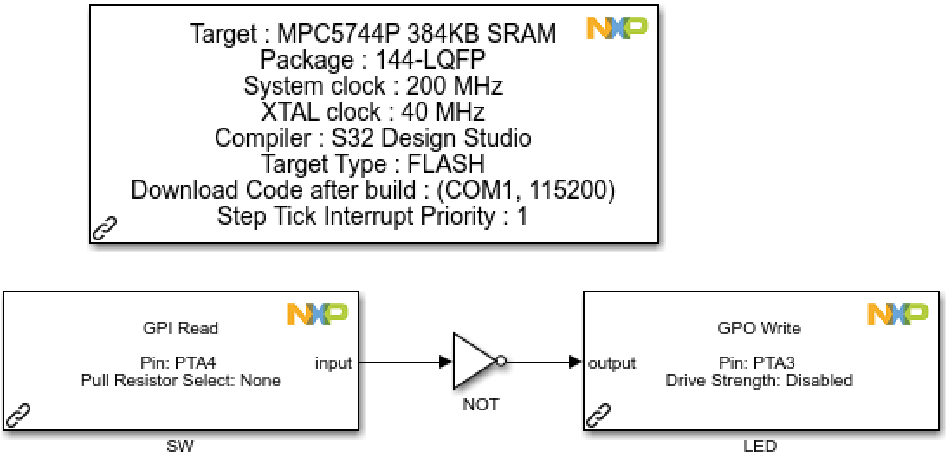


Figure 3.2.2 SIMULIINK Model of Digital Input/Output Program

When the switch is not pressed, the digital input signal from the switch is "0", and it is inverted to "1" by NOT operator. Therefore, the digital output will become "1" and the LED is ON. When we press the switch, the digital input signal from the switch is "1" and it is inverted by NOT operator. Thus, the digital output signal becomes "0", so LED is OFF.

3.2.2. Multiple Digital Inputs and Output

After we have tested the simple digital input and output, we want to develop a more complex program and apply the digital operator in it. Hence, we developed a program using XOR logic operation. This program has two inputs and one output. In this program, the LED will be ON if only one switch is pressed. Otherwise, the LED will be OFF.

Using the concept from this program, we can develop a program for the directional control of a motor. In directional control of the motor, two signals from a controller are needed to change the direction of the motor (one for clockwise direction and another for counterclockwise direction). If two signals from the controller are transmitted at the same time, the motor will be damaged. Thus, we must limit the controller to send only one signal at a time.

SW 1	SW 2	LED
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.2.2 Truth Table of Multiple Digital Inputs/Outputs

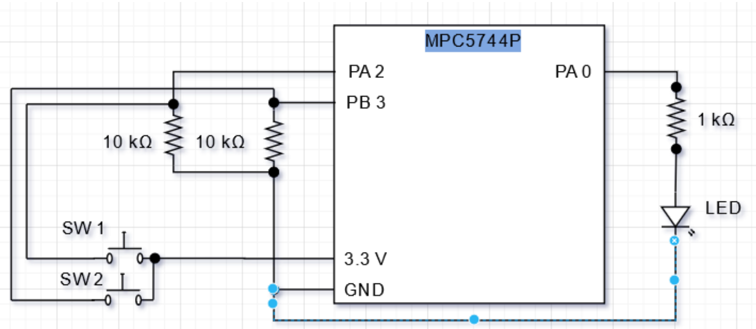


Figure 3.2.3 Circuit diagram of Multiple Digital Inputs/Outputs Program

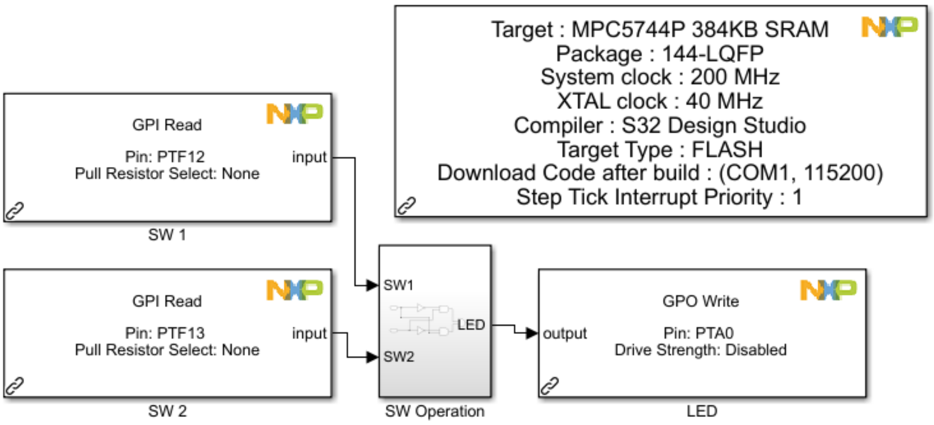


Figure 3.2.4 SIMULINK Model of Multiple Digital Inputs/Outputs Program

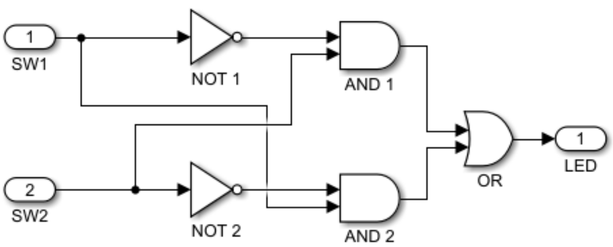


Figure 3.2.5 Switches Operations of Multiple Digital Inputs/Outputs Program

When only switch 1 is pressed, the digital input signal from switch 1 and switch 2 will be “1” and “0” respectively and the XOR operator will send the signal “1” to the digital output pin. Therefore, LED will be ON. The same concept is applied for pressing only switch 2. When both switches are pressed or not, the XOR operator will send the signal “0” as a digital output to the pin. Thus, LED will be OFF.

3.2.3. Analog Input and Output (Intensity of LED Control)

After we had done the program for digital input and output, we developed a program for analog inputs and outputs. In this program, we will use pulse width modulation to control the brightness of an LED by using a potentiometer. A potentiometer is used for analog input, and onboard LED and external LED are used for analog and digital outputs. Both LEDs were connected as the active low connection. When the output pin is high, LEDs will be OFF. By increasing the value of the potentiometer, the brightness of LEDs will be increasing. The potentiometer value is 12 bits, so we need to multiply with the gain because the output of the pin is 0 to 1. Using this concept, we can develop a program to control the speed of a motor.

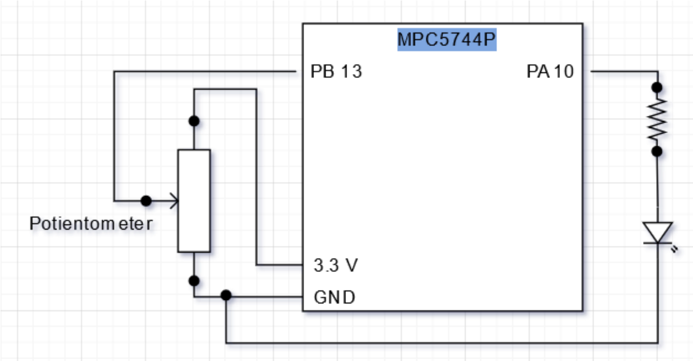


Figure 3.2.6 Circuit diagram of Analog Input/Output Program

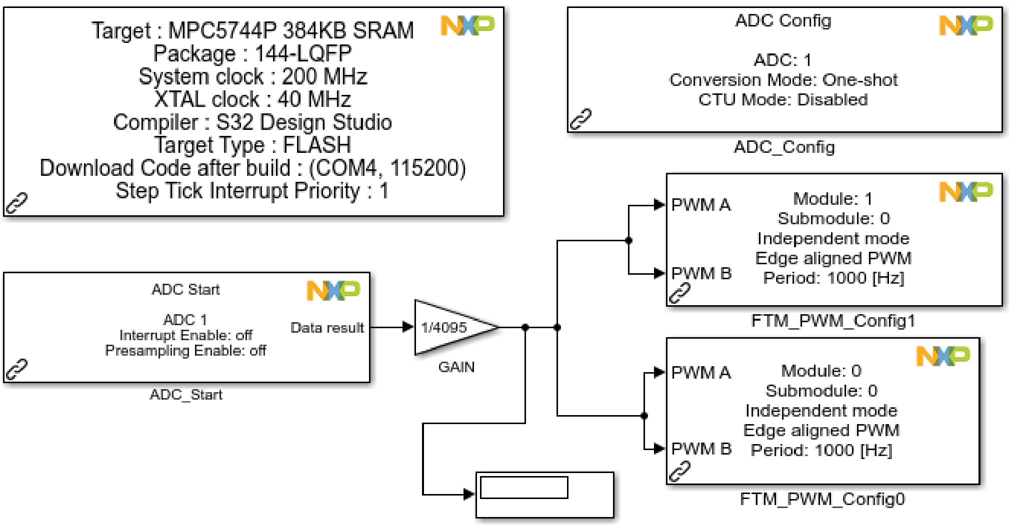


Figure 3.2.7 SIMULINK Model of Analog Input/Output Program

3.2.4. 7-Segment Display

For an application of digital inputs and outputs, we made a 7-segments display program. In this program, 7-segments will display 0 – 9 repeatedly if "START" is pressed. If "Pause" is pressed, 7-segments will stop running and pause at the current number. If "Pause" is pressed again, 7-segments will continue running from the current number. If "STOP" is pressed, 7-segments will be reset and display "0".

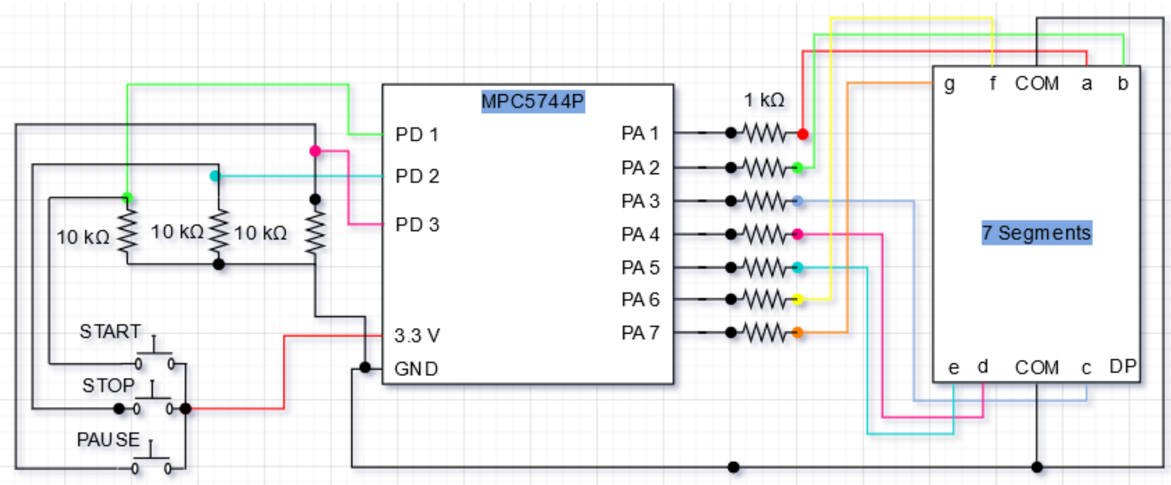


Figure 3.2.8 Circuit diagram of 7-Segment Display Program

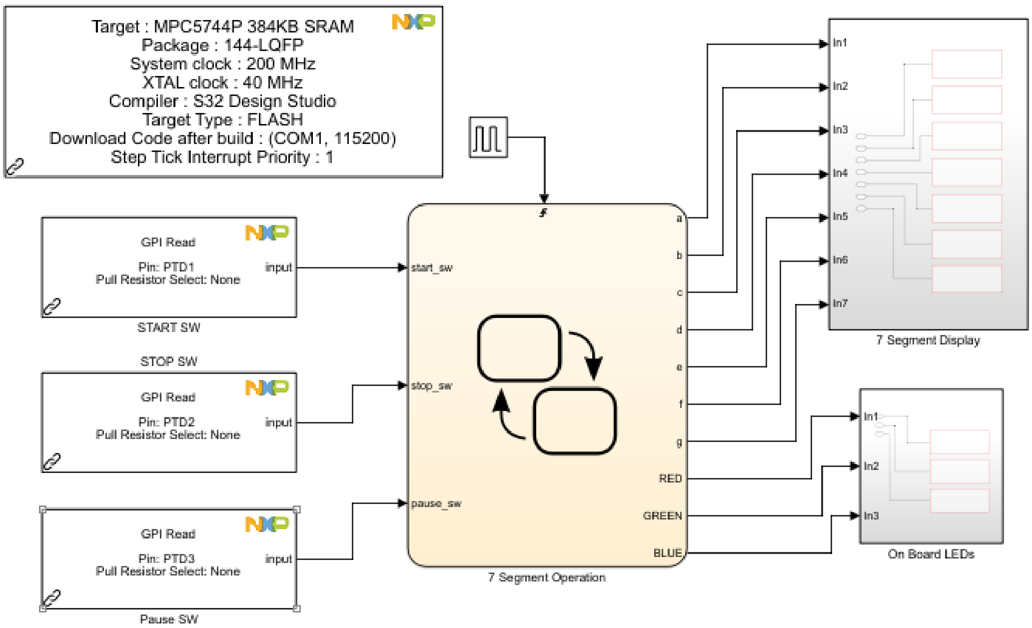


Figure 3.2.9 SIMULINK Model of 7-segment Display Program

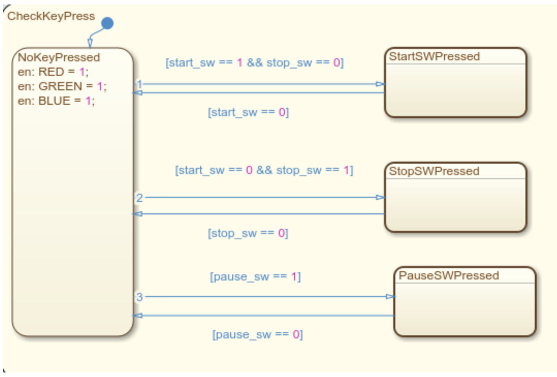


Figure 3.2.10 Switch operation Block of 7-segment Display Program

Figure 3.2.10 shows how the switches are operated.

3.3. Timer and Alarm

3.3.1. Introduction to Timer and Alarm: Timer

This program is designed to let students be able to turn ON/OFF the devices using the timer. In it the desired value is to be set (in seconds). When the "start switch" is pressed, the timer will start counting. When timer counting = timer set value, the output will be "1" and it will turn ON/OFF the device (example: motor). When the "reset switch" is pressed, the timer will reset to "0".

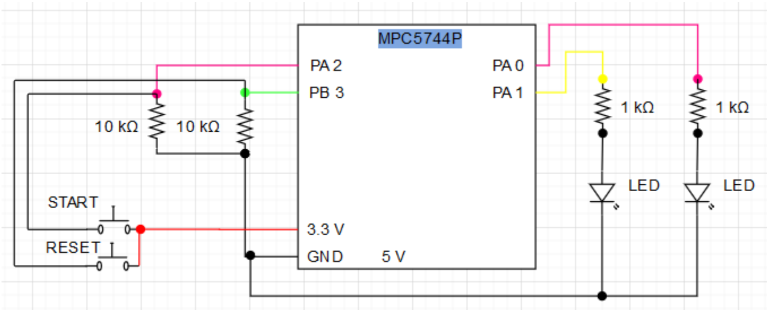


Figure 3.3.1 Circuit diagram of Timer Program

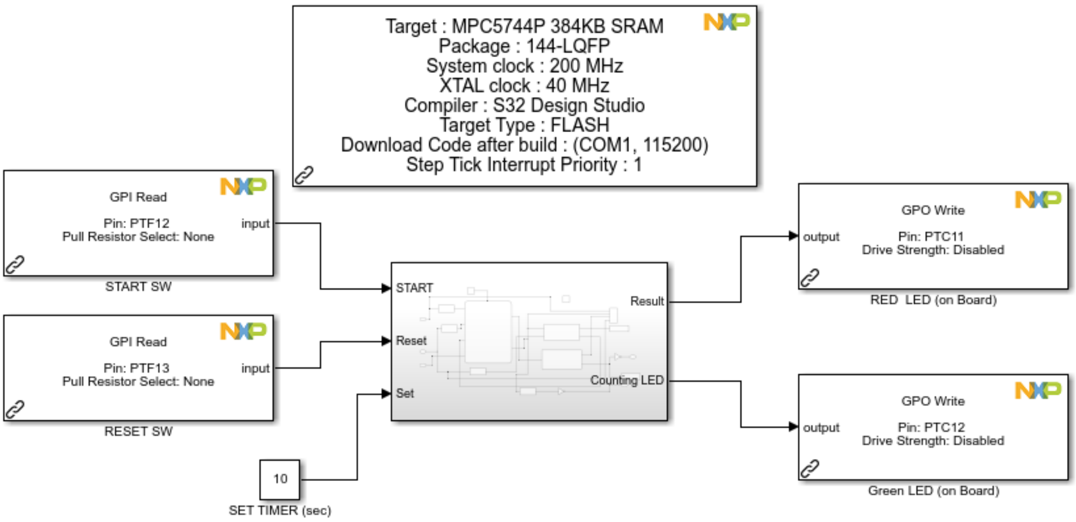


Figure 3.3.2 SIMULINK Model of Timer Program

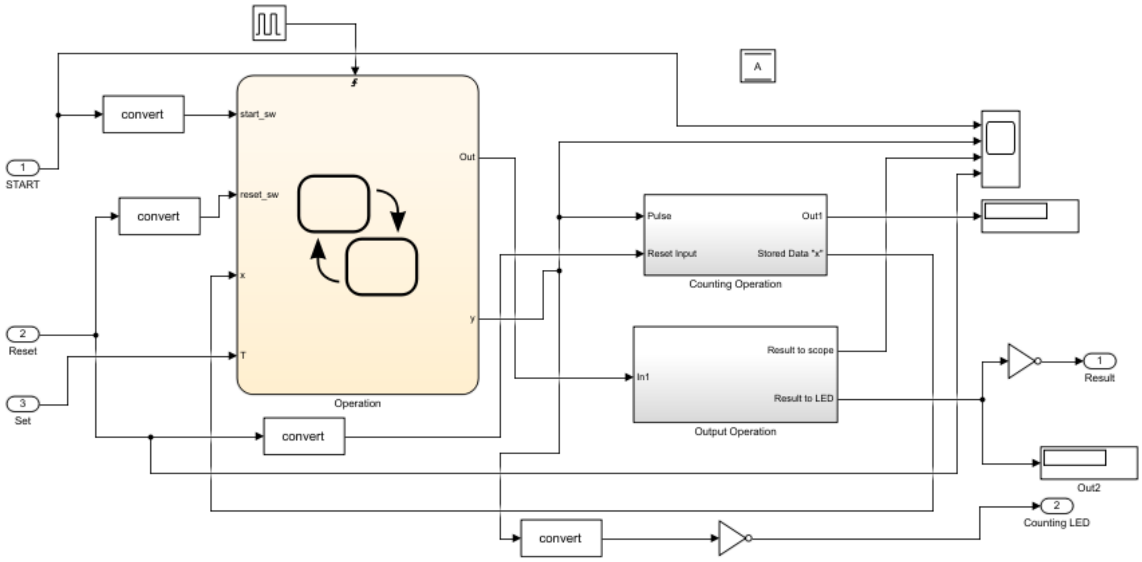


Figure 3.3.3 Operation Blocks of Timer Program

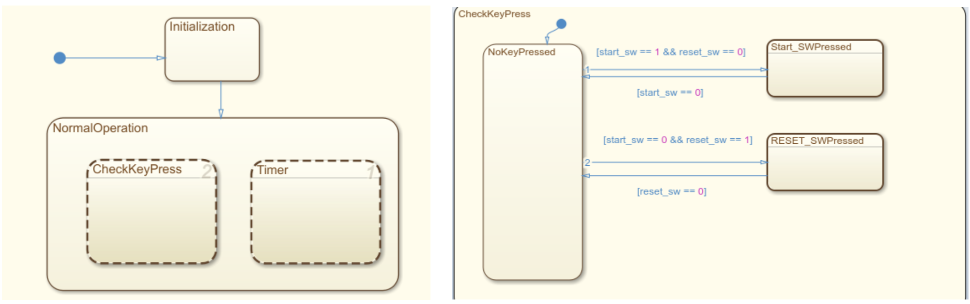


Figure 3.3.4 State Flow Chart of Timer Program

Figure 3.3.4 shows how start and reset switches work.

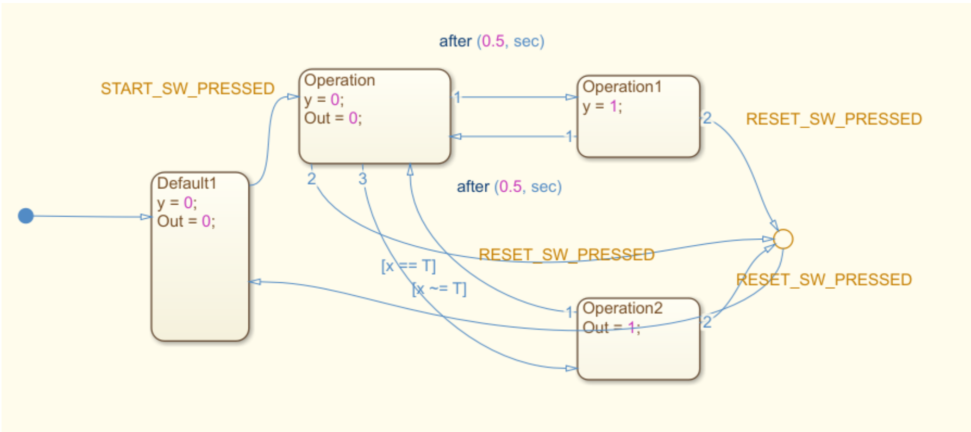


Figure 3.3.5 Sub - State Flow Chart Operation of Timer Program

Figure 3.3.5 shows how the timer operation produces timer signal and how the output signal is produced. If start switch is pressed, timer signal will periodically be sent

“0” and “1” for 0.5 sec each. If reset switch is pressed, timer signal will send signal “0”. Output signal will become “1” when timer value, “x” is set. Otherwise, output is “0”.

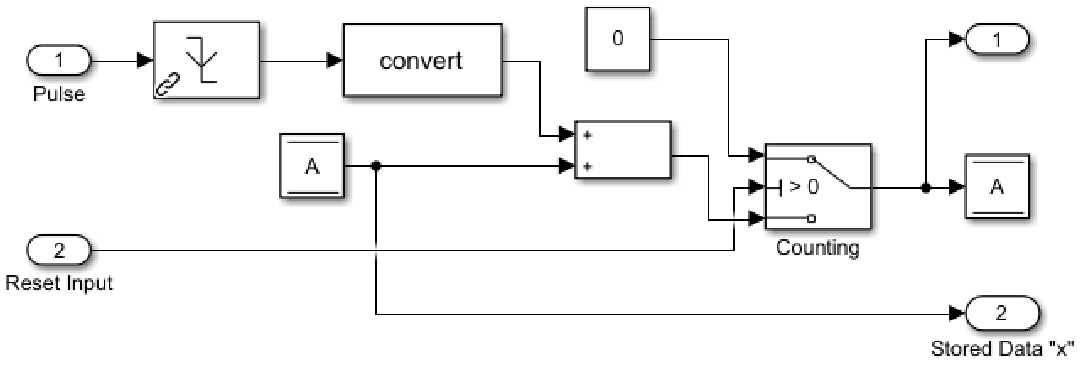


Figure 3.3.6 Counting Operation of Timer Program

In *figure 3.3.6*, the signal from timer operation block is counted at every falling edge as “1” which means 1 seconds and every counted value “1” will be added until it reaches to set value of timer. Meanwhile, the current counting value of timer, “x” is sent back to operation block to compare if it is equal to the set value of timer.

3.3.2. Introduction to Timer and Alarm: Alarm

In this topic, the alarm system is programmed using only a state flow chart. This program is designed for safety of the motor over high temperature. In it, the desired temperature value must be set first. The measured value from the temperature sensor will be then compared with the set value of temperature. If measured value ≥ set value, alarm LED will be ON (blinking with a period of 1 sec and 50% duty cycle)

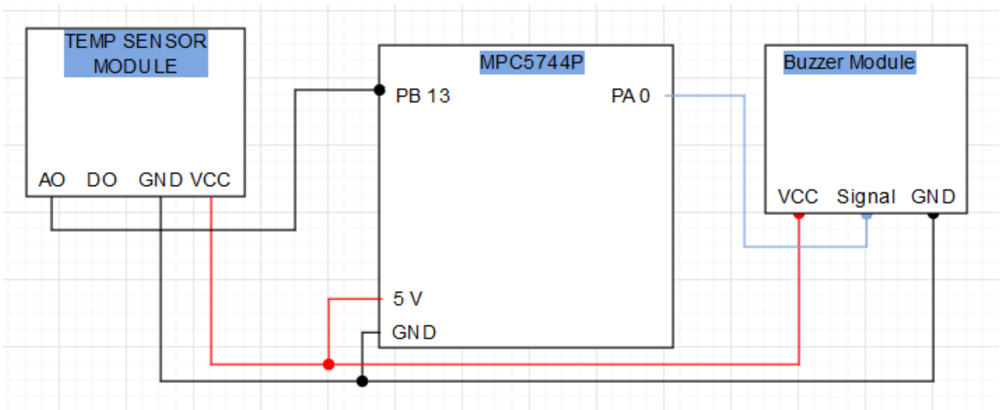


Figure 3.3.7 Circuit diagram of Alarm Program

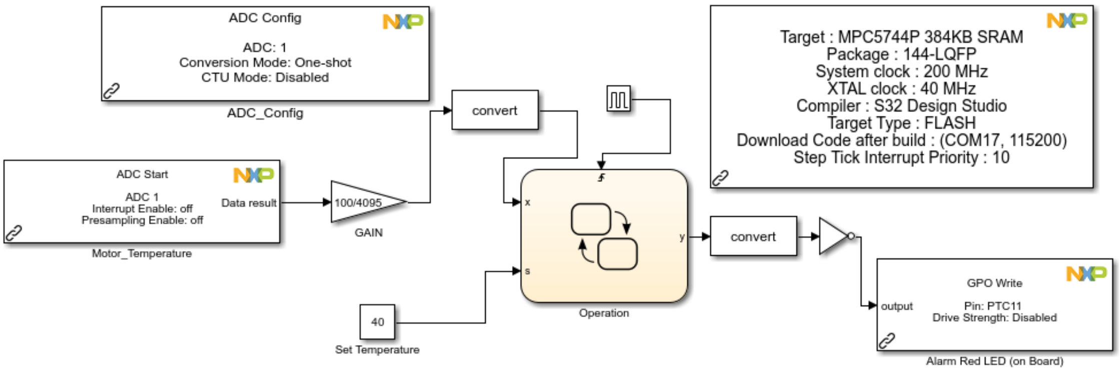


Figure 3.3.8 SIMULINK Model of Alarm Program

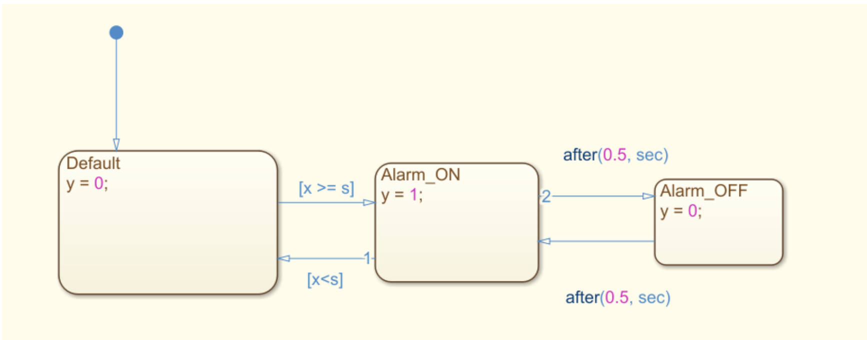


Figure 3.3.9 State Flow of Alarm Program

Figure 3.3.9 shows how the alarm signal is sent to output. When measured temperature value, “x” is less than set temperature value, “s”, output signal will be “0”. Otherwise, output signal will be repeating “1” and “0” for 0.5 seconds each until “x” is less than “s”.

3.3.3. Application of Timer and Alarm: Safety System for Motor Temperature

In this topic, the alarm system is programmed using the blocks from the NXP-MBD toolbox. Timer operation program is the same as the above timer program. Desired values of temperature and timer must be set first. The process will be the same with the alarm program above. Alarm operation controls the start and reset of the timer. Once the alarm is ON, the timer will be activated. If the measured value of temperature falls below the set temperature before the timer reaches the set value, the motor will keep running. Otherwise, the motor will stop running once the timer is up.

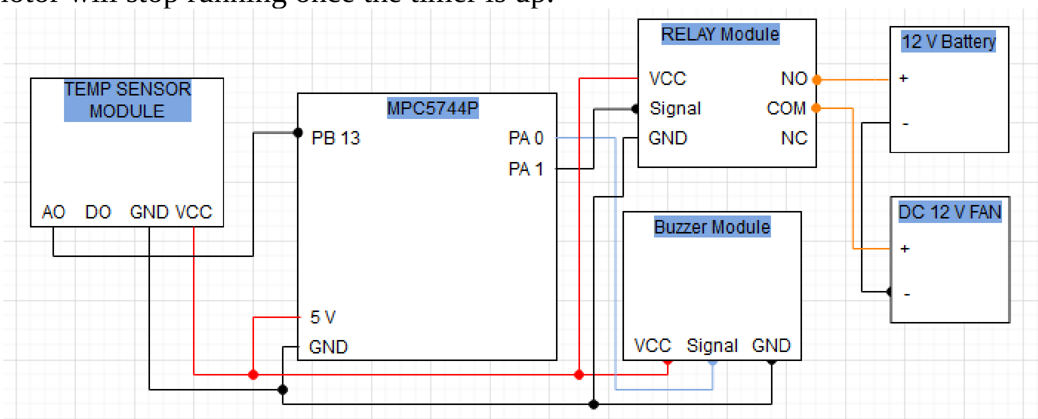


Figure 3.3.10 Safety System for Motor Over Temperature Circuit Diagram

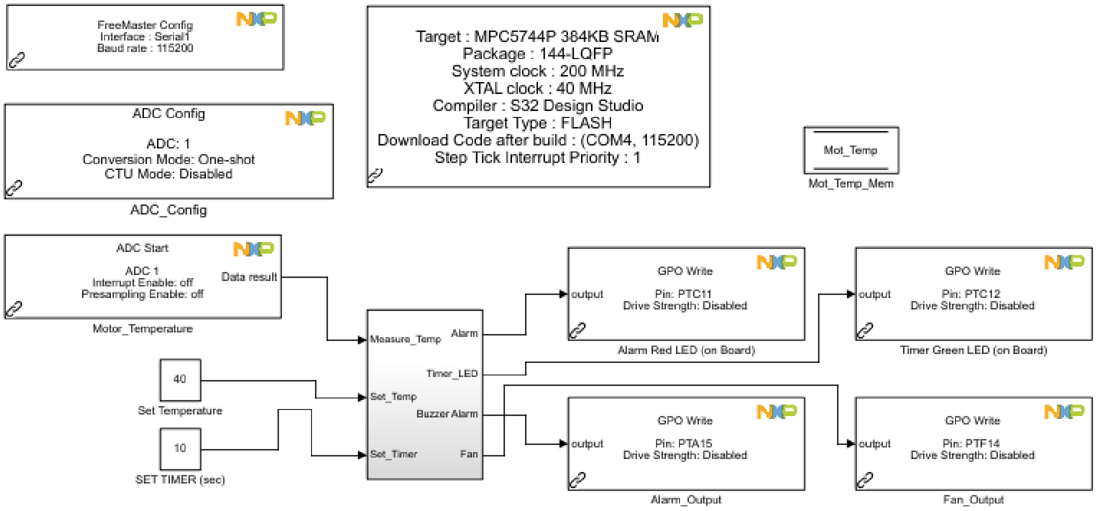


Figure 3.3.11 SIMULINK Model of Safety System for Motor Over Temperature

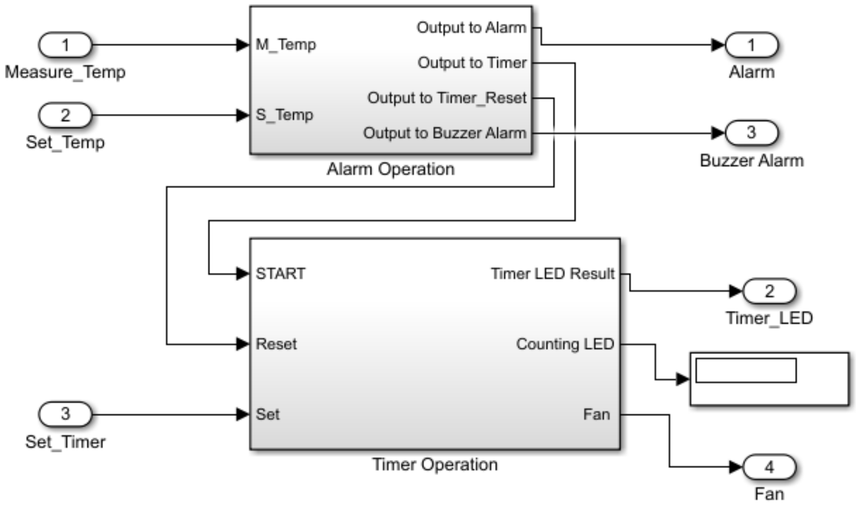


Figure 3.3.12 Operation Blocks of Safety System for Motor Over Temperature

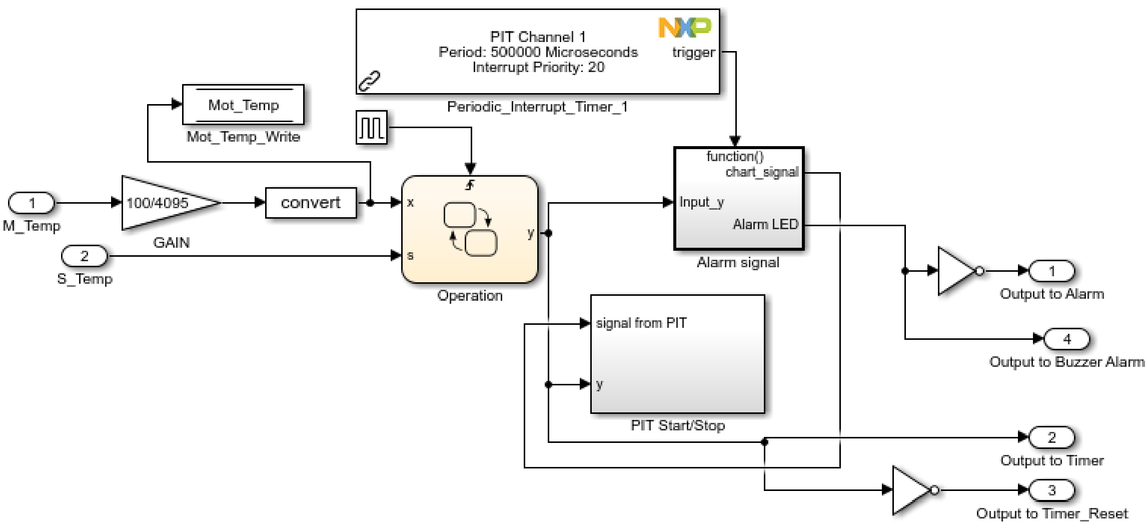


Figure 3.3.13 Alarm Operation System of Safety System for Motor Over Temperature

In *figure 3.3.13*, alarm signal is sent by operation block as “0” or “1”. PIT block is used to generate the pulse signal for alarm.

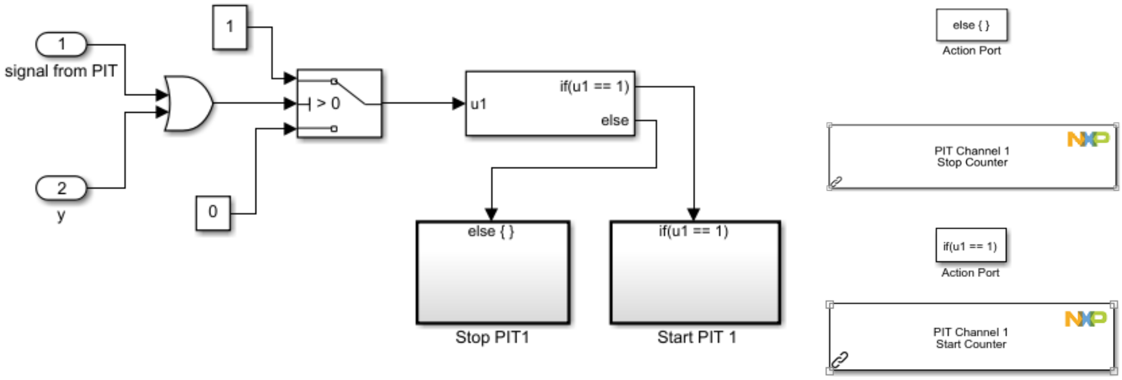


Figure 3.3.14 Periodic Interrupt Timer (PIT) Start/Stop of Safety System for Motor Over Temperature

In *figure 3.3.14*, the signal from operation block, “y” and signal from PIT block are operated in OR operation and the output signal is used as a switch to start or stop PIT block operation.

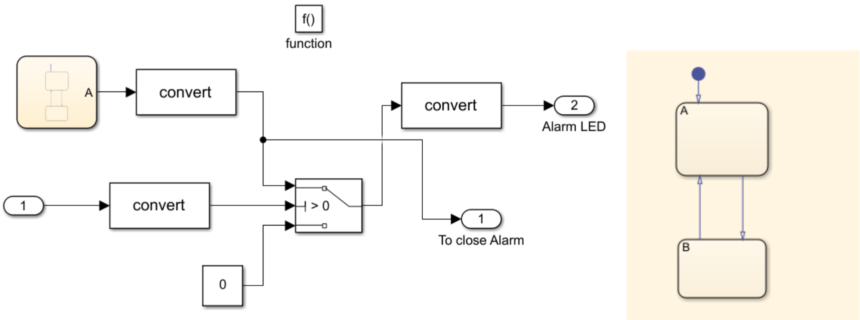


Figure 3.3.15 Alarm Signal Operation of Safety System for Motor Over Temperature

In *figure 3.3.15*, the chart sends the output signal, “A” as “0” or “1”. Meanwhile the signal from the chart is sent to “PIT Start/Stop” block. If the signal from operation block, “y” is 1, the alarm output will repeat “0” and “1” for 0.5 sec each until the temperature drops below set temperature value. Otherwise, alarm output is ‘0” which means “OFF”.

3.4. DC Stepper Motor Control: Speed and Direction Control

In this program, the speed of the motor will be adjusted by an accelerator (potentiometer) and an ultrasonic sensor will be used to sense the object (application: in front of a car). The speed will be reduced if the ultrasonic sensor detects an object. Depending on the distance, the speed will be reduced with the specified percentage.

Signal of the distance will be used as “min (cm) = 0% to max (cm) = 100 %”

Signal of distance (less than or equal to)	Speed reduction
100 %	50 %
85 %	40 %
70 %	30 %
60 %	20 %
50 %	10 %
45 %	0 %

Table 3.4.1 Speed Reduction Depending on the Signal of Distance from Ultrasonic Sensor

By pressing CW and CCW switch, direction of the motor is controlled. XOR function is used to prevent short circuit in motor driver.

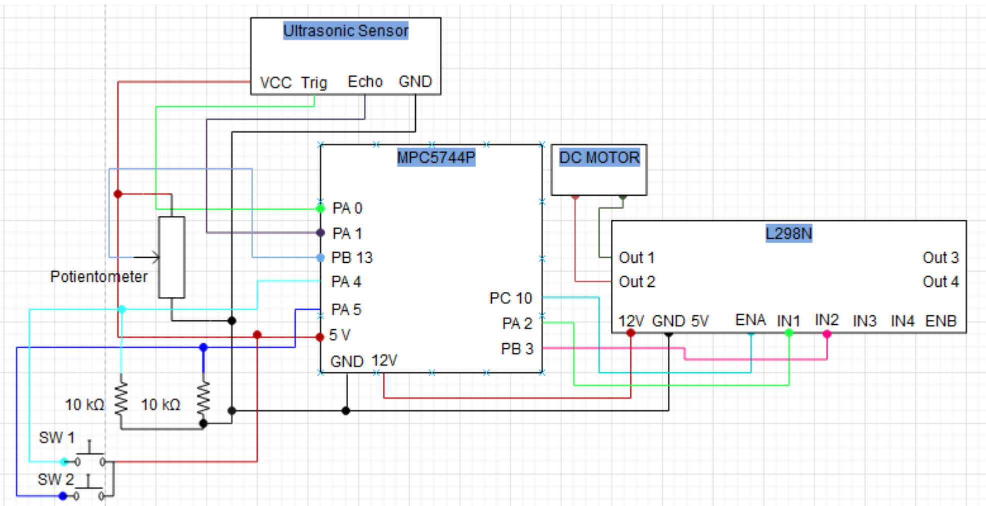


Figure 3.4.1 DC Motor Control Circuit Diagram

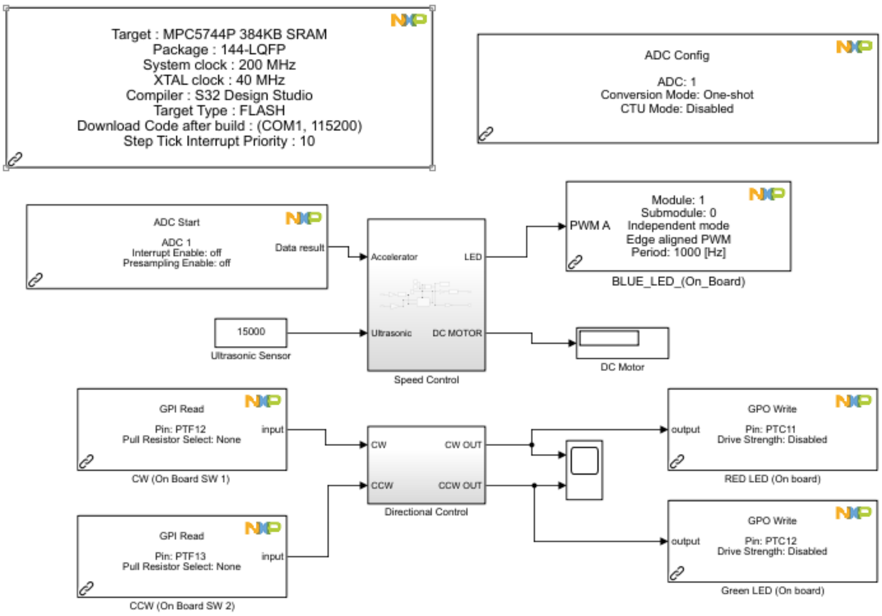


Figure 3.4.2 SIMULINK Model of DC Motor Control

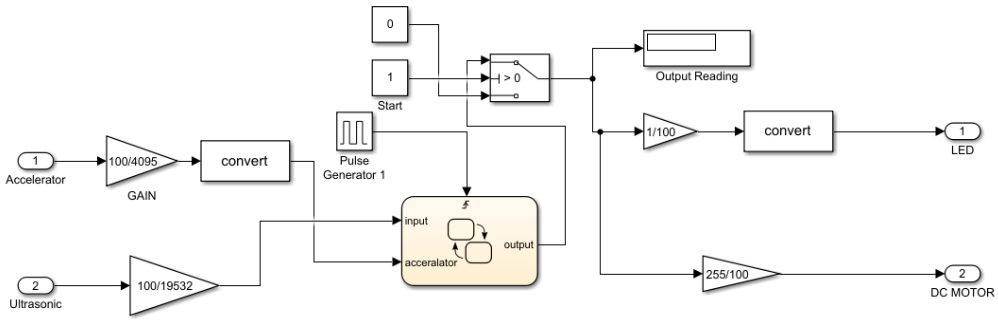


Figure 3.4.3 Speed Control Operation of DC Motor Control

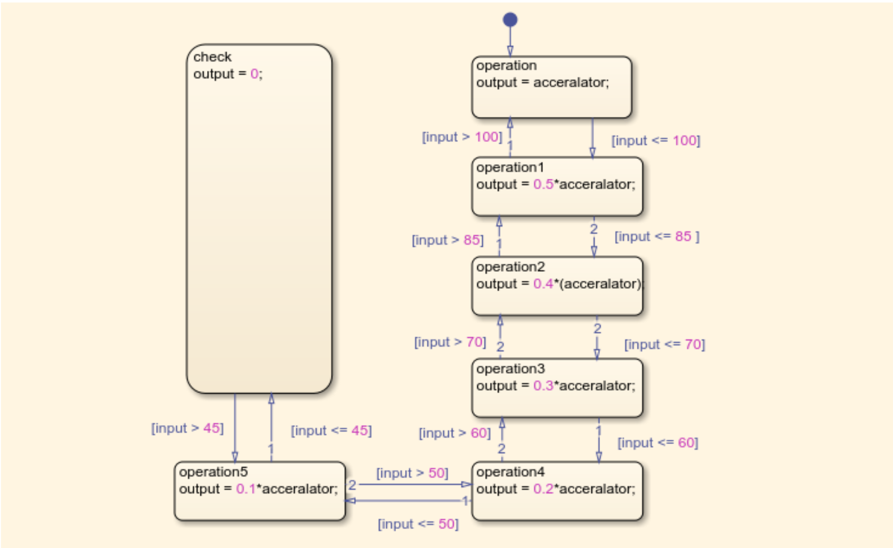


Figure 3.4.4 State Flow Chart of DC motor speed Control

Figure 3.4.4 shows how speed reduction is operated depending on the data from ultrasonic sensor. The output speed will be same with the speed of “accelerator” which is the data from potentiometer. Depending on the percentage of the data from ultrasonic sensor, “input”, the output speed will decrease or increase. The maximum and minimum percentage of output speed are 100 and 0 respectively.

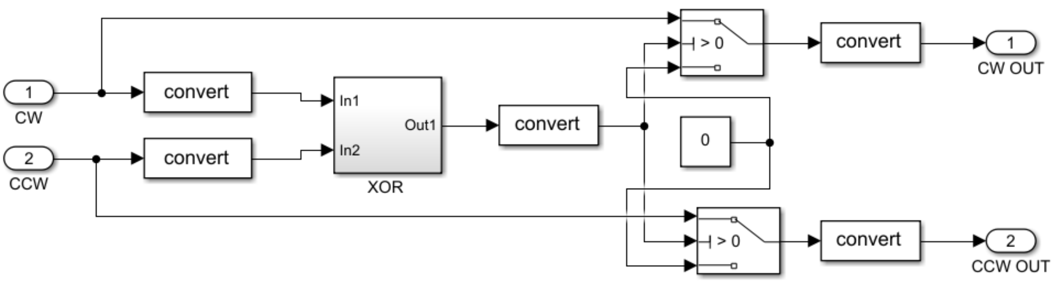


Figure 3.4.5 Directional Control Operation of DC Motor Control

Figure 3.4.5 shows how direction of the motor is controlled. CW and CCW switches control the motor to rotate clockwise and counterclockwise direction, respectively. As the advantage of XOR function, both CW and CCW switches are not allowed to press at the same time.

3.5. Control Area Network (CAN) Communication

3.5.1. CAN Communication Using GY8507

The host computer will connect to VCU and GY8507 CAN USB adapter for monitoring the transmit and receive data between those devices.

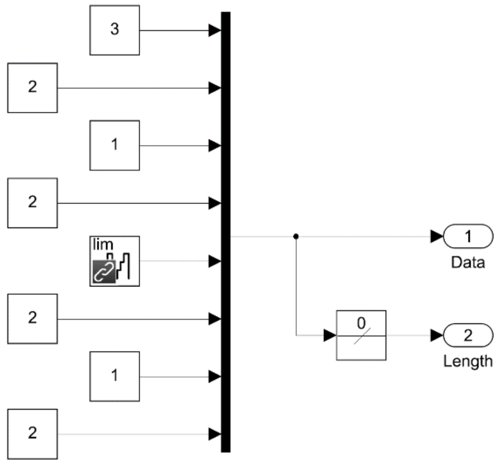


Figure 3.5.1 CAN Frame

We generate the CAN message with continuously counting by using the counter limit to change the control bit every second. The counter limit was set to 8 bits of counting.

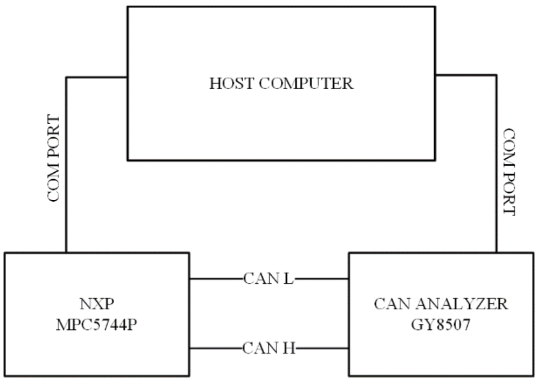


Figure 3.5.2 Testbench Diagram

The controller will keep sending the data continuously with ID:0x3FF. The baud rate was set to be the standard of electric vehicle CAN communication of 250 kbps. The CAN analyzer able to monitor the transmit data of the controller. Moreover, host computer will send the message through CAN analyzer to the controller by having the receive ID:0x3FE. Whenever the host computer sends the data to controller the on-board LED will turn on without interfere the transmit message.

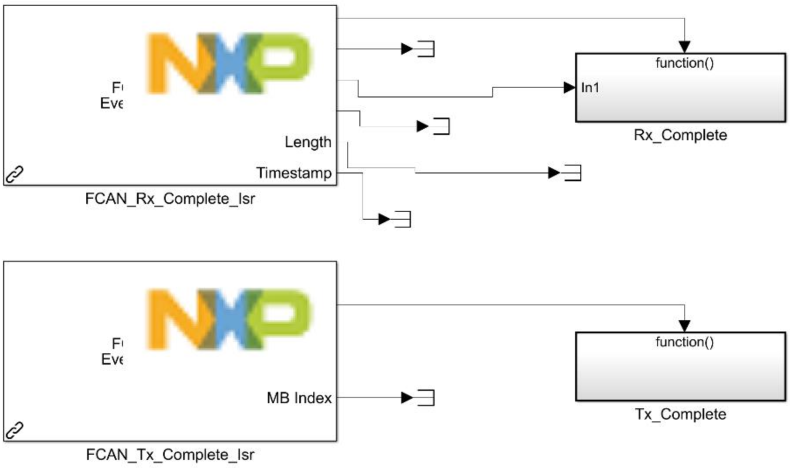
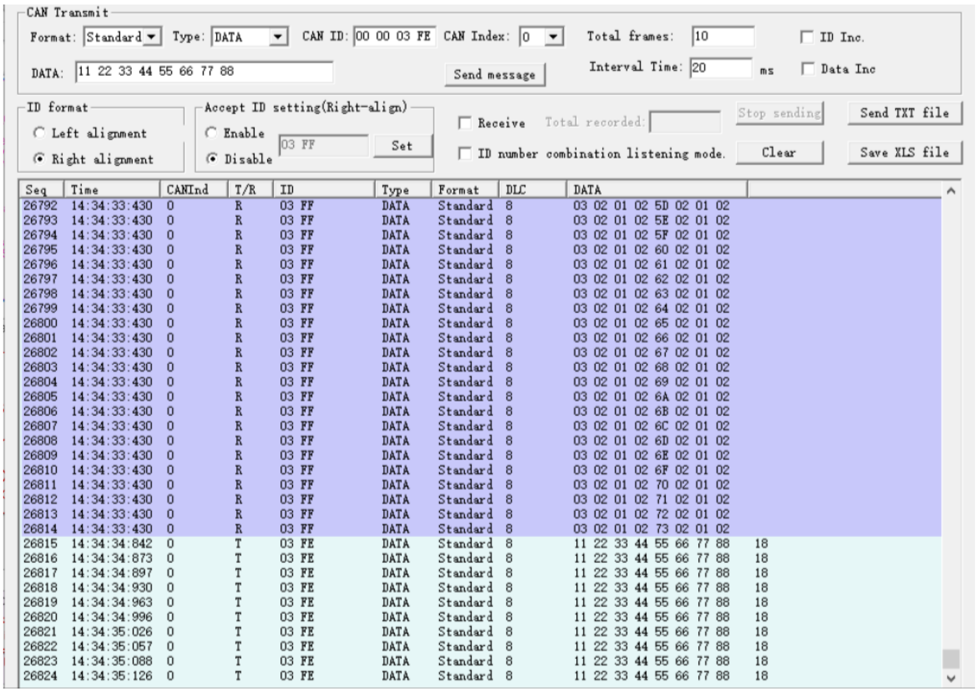


Figure 3.5.3 SIMULINK Model for VCU

The monitoring in CAN analyzer shows us the result that we expected to get with the basic understanding of CAN. To receive the data from the controller or transmit the data to controller, we need to select the same baud rate as we configurate in the SIMULINK VCU toolbox. If the baud rate is not identical, we could not monitor and transmit the data into VCU.



Seq	Time	CANInd	T/R	ID	Type	Format	DLC	DATA	
26792	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 5D 02 01 02	
26793	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 5E 02 01 02	
26794	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 5F 02 01 02	
26795	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 60 02 01 02	
26796	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 61 02 01 02	
26797	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 62 02 01 02	
26798	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 63 02 01 02	
26799	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 64 02 01 02	
26800	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 65 02 01 02	
26801	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 66 02 01 02	
26802	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 67 02 01 02	
26803	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 68 02 01 02	
26804	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 69 02 01 02	
26805	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6A 02 01 02	
26806	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6B 02 01 02	
26807	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6C 02 01 02	
26808	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6D 02 01 02	
26809	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6E 02 01 02	
26810	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 6F 02 01 02	
26811	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 70 02 01 02	
26812	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 71 02 01 02	
26813	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 72 02 01 02	
26814	14:34:33:430	0	R	03 FF	DATA	Standard	8	03 02 01 02 73 02 01 02	
26815	14:34:34:842	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26816	14:34:34:873	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26817	14:34:34:897	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26818	14:34:34:930	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26819	14:34:34:963	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26820	14:34:34:996	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26821	14:34:35:026	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26822	14:34:35:057	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26823	14:34:35:088	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18
26824	14:34:35:126	0	T	03 FE	DATA	Standard	8	11 22 33 44 55 66 77 88	18

Figure 3.5.4 Monitoring CAN Bus (The purple color is the transmit data from the VCU whereas green is the transmit data from host computer into the VCU)

3.5.2. CAN Communication between Microcontrollers

The CAN communication between microcontrollers is the model for communication for transmit, receive and exchange the data. To perform the CAN communication, microcontrollers must have the CAN port or CAN module to be able to connect with other. In the practical applications such as EV system, all of the electronics equipment has the CAN transceiver IC that allow them to communicate to the controller in the EV.

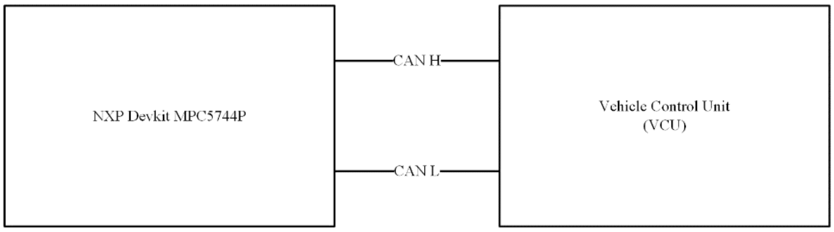


Figure 3.5.5 CAN Communication Diagram between Two Microcontrollers

In this program, the SIMULINK model of CAN communication in the EV system has simulate and testing between NXP Devkit MPC5744P and VCU.

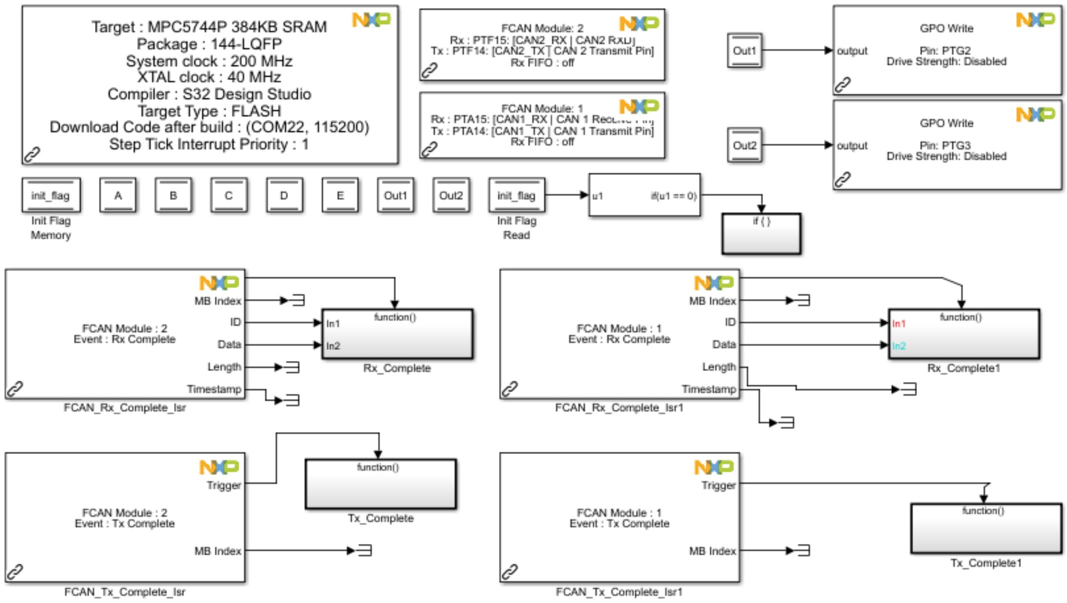


Figure 3.5.6 SIMULINK Model of CAN Communication between Two Microcontroller

The purpose of this program is to check the communication of microcontrollers via CAN bus whether it is receiving and transmitting the right CAN data that generate from each of the controller. The LED turns on whenever the each of the microcontroller receive the CAN frames from each other.

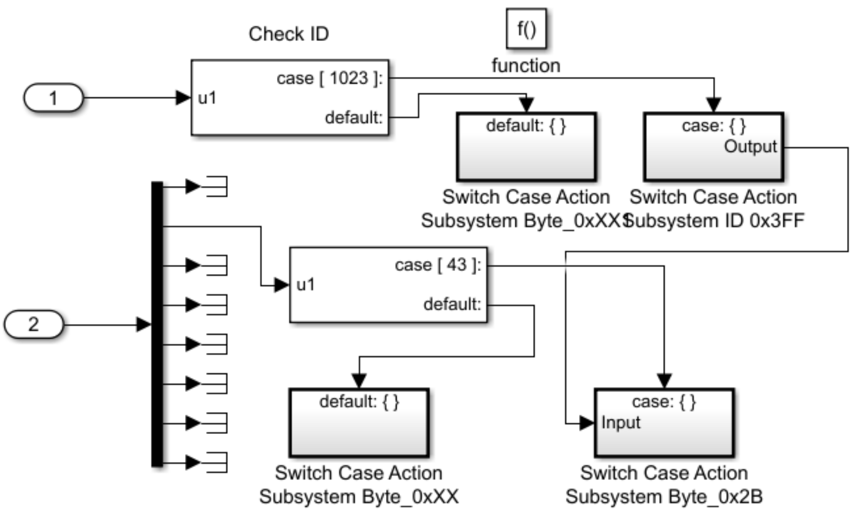


Figure 3.5.7 FlexCAN Receive

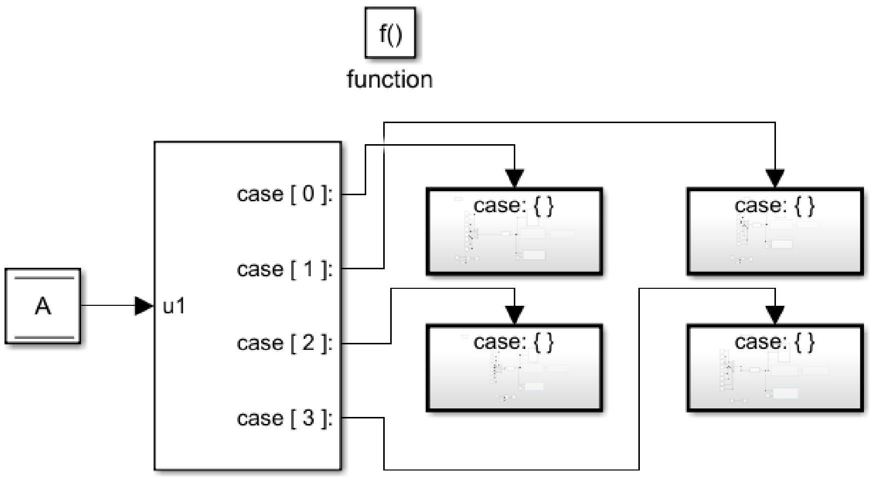


Figure 3.5.8 FlexCAN Transmit

Each of microcontrollers define their FlexCAN ID differently, if the ID is the same or invalid the microcontroller cannot communicate with each other. The CAN message transmit with four differences data which generated. The message will transmit each case one by one with increment of one number of cases. FlexCAN receive will light up the LED and the CAN analyzer used as monitoring the transmitting and receiving data. FlexCAN receive will check the CAN frame data with cases condition, if the data is incorrect the LED will not turn on unless the receiving data is correct.

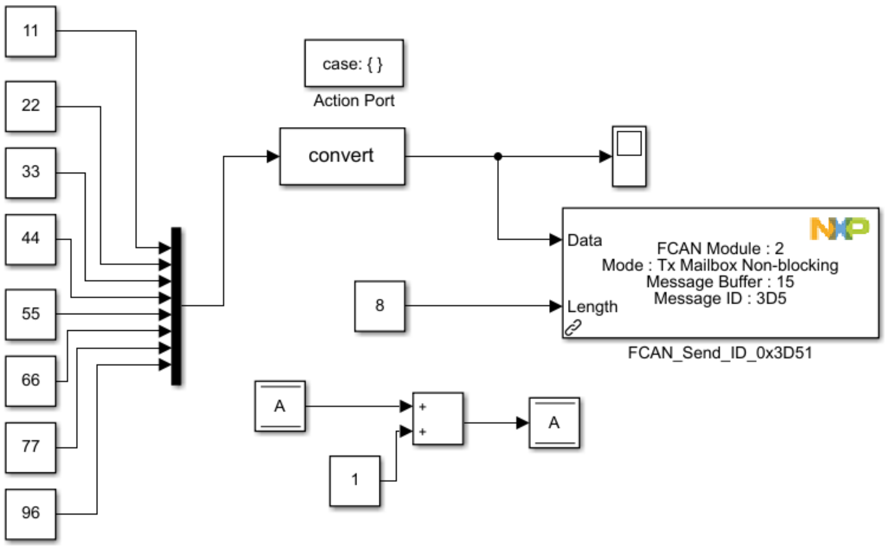


Figure 3.5.9 CAN Frame

The CAN message generated by using multiplexer in each of the cases in the transmitting block. The SIMULINK will show the real-time transmit and receive the data as well.

3.6. Brush DC Motor Control

In this program, the speed of a brush DC motor is controlled by armature control method. In addition, the speed of the motor is maintained by the feedback control system in which PID controller and a speed sensor are used. The speed sensor is used to measure the actual speed of the motor and we compare the value of desired speed and measured value from the sensor.

For frequency response calculation to find the gains of the PID controller, we choose maximum overshoot (Mp) as 1% and rise time as 0.1 sec. The calculation and equations are the same as described in theory “2.4.3”.

Script – file in MATLAB Editor:

```
% to find damp and neutral frequency
Mp = 0.01;
Damp_ratio = sqrt(1/(1+(pi/log(Mp))^2));
tr = 0.1;
beta = atan2d(sqrt(1-Damp_ratio^2),Damp_ratio);
beta_rad = beta*(pi/180);
wn = (pi-beta_rad)/(tr*sqrt(1-Damp_ratio^2));

% parameter of DC motor
Ra = 25;
L = 0.336;
Jm = 0.00295;
Bm = 0.00583;
Kt = 7.17;
Ke = 7.17;

A = L*Jm;
B = Ra*Jm +Bm*L;
C = Ra*Bm + Kt*Ke;

% to find Kd, Kd, Ki
Kd = (2*Damp_ratio*wn-B)/Kt;
Kp = (wn^2)/C*Kt;
Ki = 100;

% system transfer function
s = tf('s');
P = Kt/(A*(s^2)+B*s+C);
C = pid(Kp,Ki,Kd);
T = feedback(C*P,1);
step(T)
```

PID gains	Estimated values	Tuned values
K_p	283.1778	300
K_i	100	500
K_d	10.38	15

Table 3.6.1 PID gains values for motor speed control system

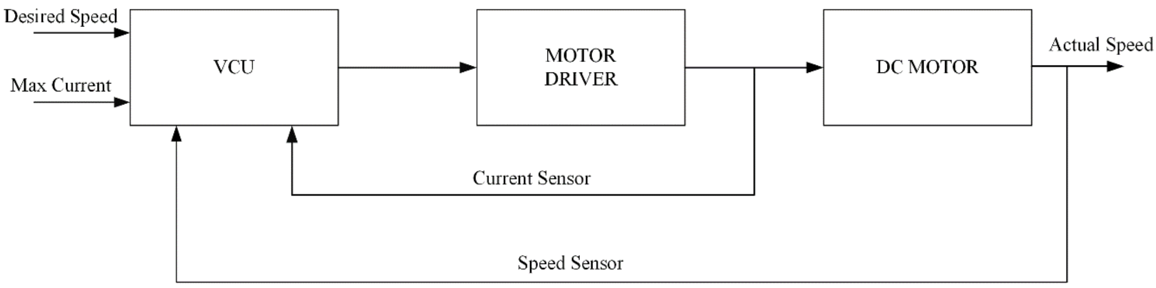


Figure 3.6.1 DC Motor Feedback Control Diagram

The overview of DC motor control using the feedback system control theory. The VCU will be used as the main controller, torque control and speed control. The desired speed will be read by VCU and calculate the error from the actual speed and send back to the motor driver to drive the motor. The current sensor used to measure the actual current and sense to VCU to calculate the error.

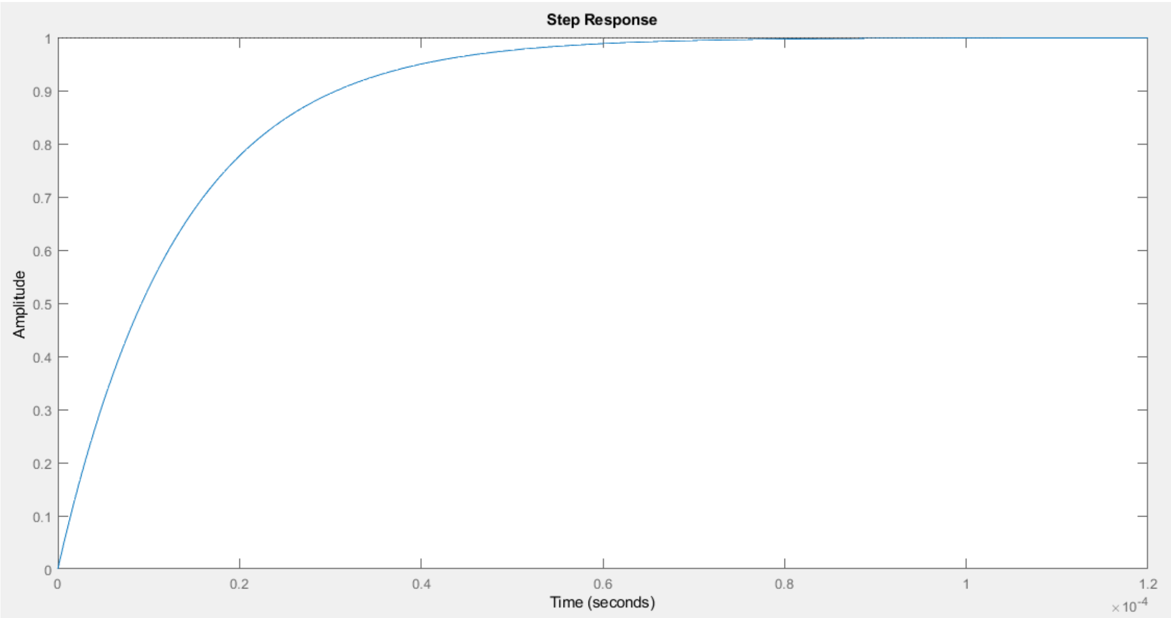


Figure 3.6.2 System response of T(s) with estimated PID gains

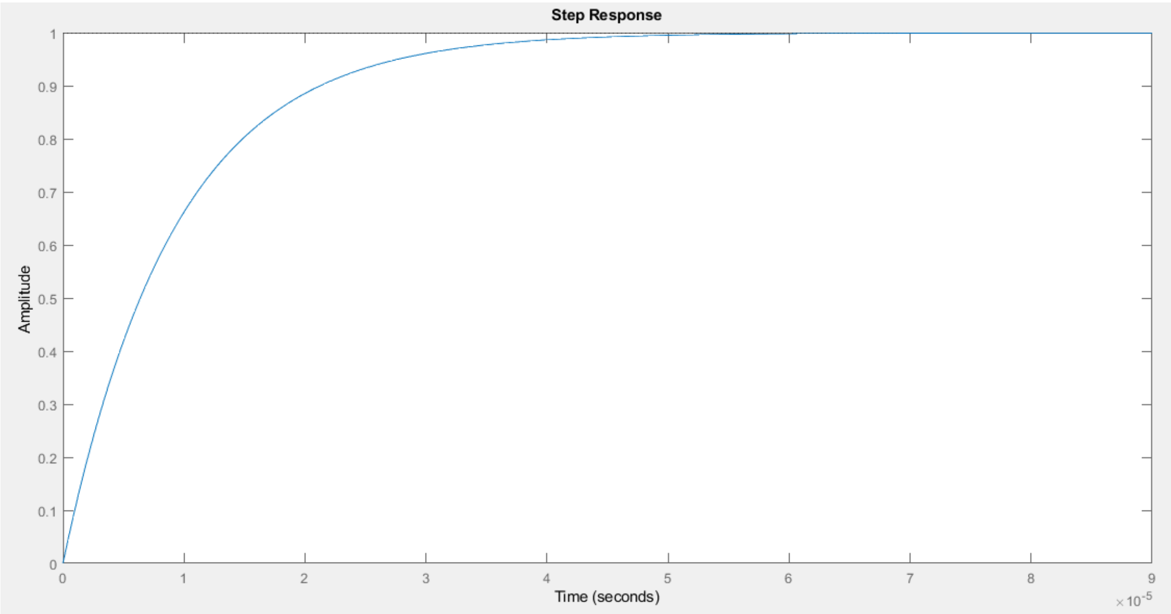


Figure 3.6.3 System response of T(s) with tuned PID gains

With tuned values of PID gains, overshoot is increased and rise time is decreased. Moreover, the system is stable at 60 μs.

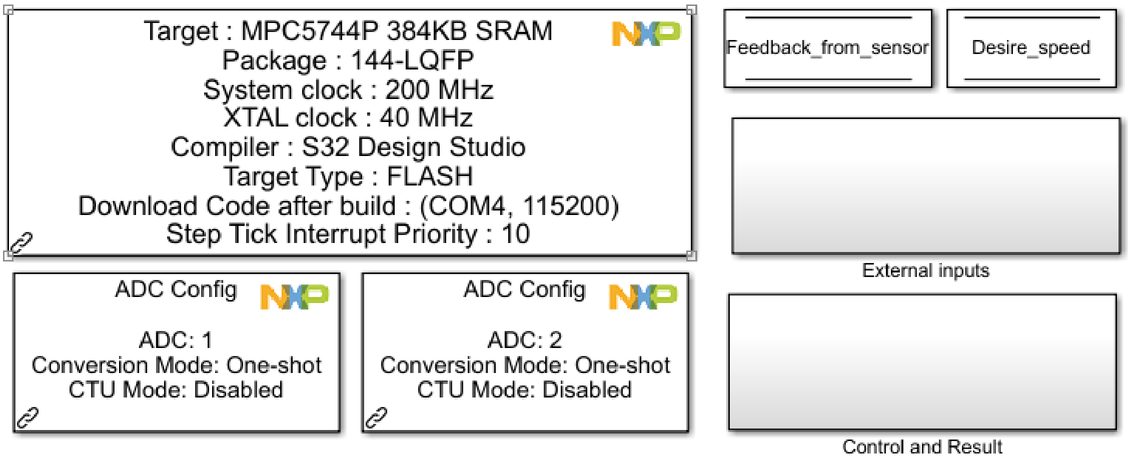


Figure 3.6.4 Simulink Model of DC motor control

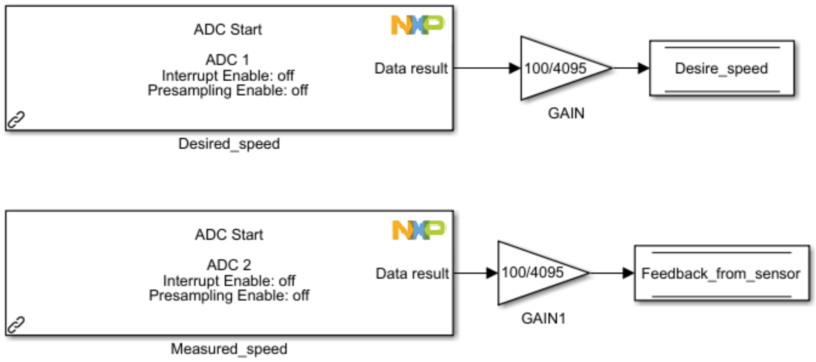


Figure 3.6.5 External Input Block of DC motor control

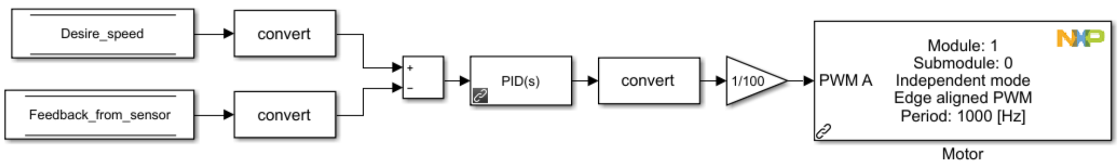


Figure 3.6.6 Control and Result Block of DC motor control

In *figure 3.6.4*, the data from desired speed and measured speed are stored in the data memory separately. In *figure 3.6.5*, the value of measured speed subtracts from the value of desired speed and it is sent to PID control block which maintain the speed of actual motor to be the same with desired speed value.

3.7. Electric Vehicle DC – DC Converter

3.7.1. CAN Protocol for EV DC – DC Converter

The protocol for DC – DC converter is using the extended frame communication with 250 kbps of baud rate. The VCU will connect directly to the DC – DC converter through the CAN port. DC – DC is controllable by using the CAN command from the VCU with its output range. The output voltage range can be controlled by m value (power supply type). The n value that represent in the CAN ID depends on the number of the DC – DC converter that connect to the VCU. The n value starts from 0.

Output Voltage Range	m value
$0\ V \leq V < 16\ V$	0

ID	Type	Period(ms)	Byte	VCU	DC-DC
0x18mnF5E5	P	200	8	Transmit	Receive
0x18nmE5F5	P	200	8	Receive	Transmit

Table 3.7.1 Extended Frame Communication

For our DC – DC converter, the output voltage range is from 10 V to 16 V. The output voltage will be set as constant voltage at 12 V. Since most of the electronics equipment in automotive are used 12 V as supply voltage.

Message Protocol on DC – DC Converter

OUTPUT	INPUT	CAN ID	Cycle (ms)
DC – DC	VCU	0x1800E5F5	200

Location	Data Name	Data
Byte0	Reserve	
Byte1	Fault alarm low byte	0: normal, 1: fault
Byte2	Fault alarm high byte	0: normal, 1: fault
Byte3	Output current low byte	0.1A/bit offset: 0 Iset = 1000, its equivalent to 100A
Byte4	Output current high byte	
Byte5	Output end voltage	m = 0; 0.2V/bit offset: 0
Byte6	Working status	0: work; 1: stop
Byte7	Shell Temperature	1V/bit offset: -40 Temp = 40, it’s equivalent to 0 ℃

Table 3.7.2 CAN Protocol of DC - DC Converter

We check on the DC – DC converter based on CAN protocol as shown in *table 3.7.2*. The CAN protocol is used to transmit the data and receive data between VCU and converter. We set the output to be at 12 V with the current depending on the number of loads which are connected to the output terminal. The DC – DC converter will transmit the data back to the VCU for operation if there is fault or abnormal conditions DC – DC will automatically switch off to avoid further damage.

Message Protocol on VCU

OUTPUT	INPUT	CAN ID	Cycle (ms)
VCU	DC – DC	0x1800F5E5	200

Location	Data Name	Data
Byte0	Reserve	
Byte1	Switch on/off control	0: Shutdown command 1: Power on command
Byte2	Set voltage low byte	0.1V/bit offset: 0; V_{set} = 120(Byte2 + Byte3*256) it is equivalent to 12V
Byte3	Set voltage high byte	
Byte4	Set current low byte	0.1A/bit offset: 0; Iset = 1000(Byte4 + Byte5*256); it is equivalent to 100A
Byte5	Set current high byte	
Byte6	Reserve	
Byte7	Reserve	

Table 3.7.3 CAN Protocol of VCU

The CAN protocol of VCU use to control the DC – DC converter with the control data of the CAN message. Without the command send to the converter, we will not get any result output at the terminal even though the input voltage being supply the on-board controller of DC – DC converter will not activate until the boot command is sent.

3.7.2. Testbench of EV DC – DC Converter

The testbench model EV DC – DC converter was conducted in Power Electronics Laboratory at Vincent Mary School of Engineering. The converter used for commercial product in EV. The actual input voltage supply is 700 Vdc from the battery source. In our model testing, we could only supply up until 400 Vdc. The input voltage supply ranges from 200 Vdc to 700 Vdc. Therefore, the laboratory can provide 200 Vdc to 400 Vdc input voltage range. The on-board controller of EV DC – DC converter needs to supply external input voltage to activate the ICs on-board. The converter has procedure to power on and power of. In the testbench diagram *figure 3.7.1*, describe the connection procedure for testing converter. The commercial product testing needed to have safety protection in the testing procedure as well as the operation manual instruction before applying any command or input high voltage.

To boot the converter, the auxiliary power must supply on then the activation signal turns on later. The activation signal allows the host computer to communicate with the converter. The Vcc activation, activate the CAN transceiver IC which it let the host computer receive the data or transmit the data into converter. The switch turns on to supply the high input voltage in then the CAN command from the host computer transmit to CAN receiver on the converter to turn on the relay at the input voltage side.

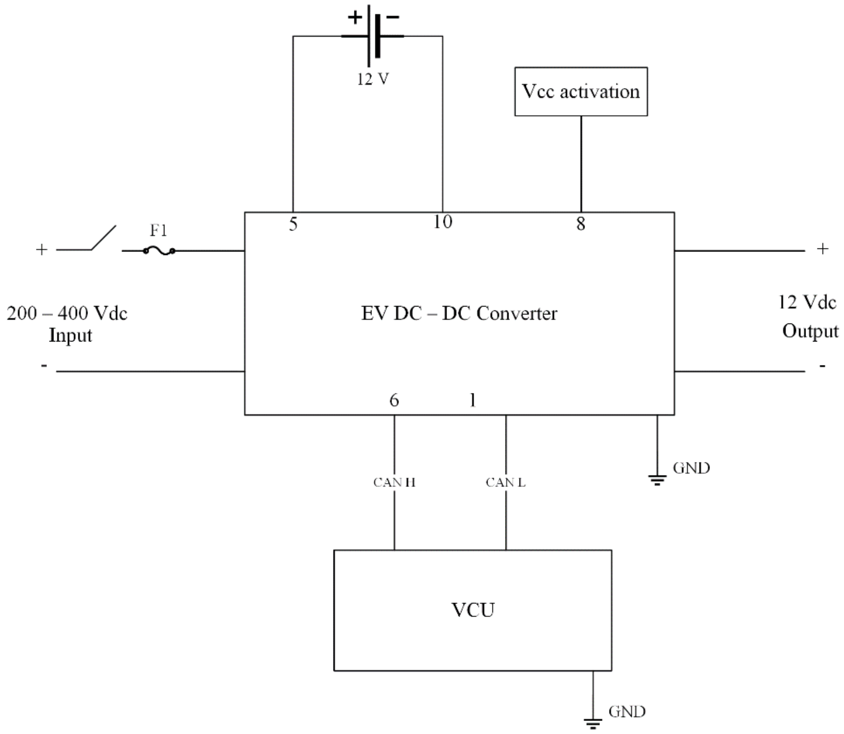


Figure 3.7.1 EV DC - DC Converter Testbench Diagram

The shutdown operation of the converter must follow the procedure on the operation manual sheet. The shutdown sequence is reverse of boot operation. The CAN command transmit the power off data to the converter. Then, disconnect the high input voltage supply from the converter.

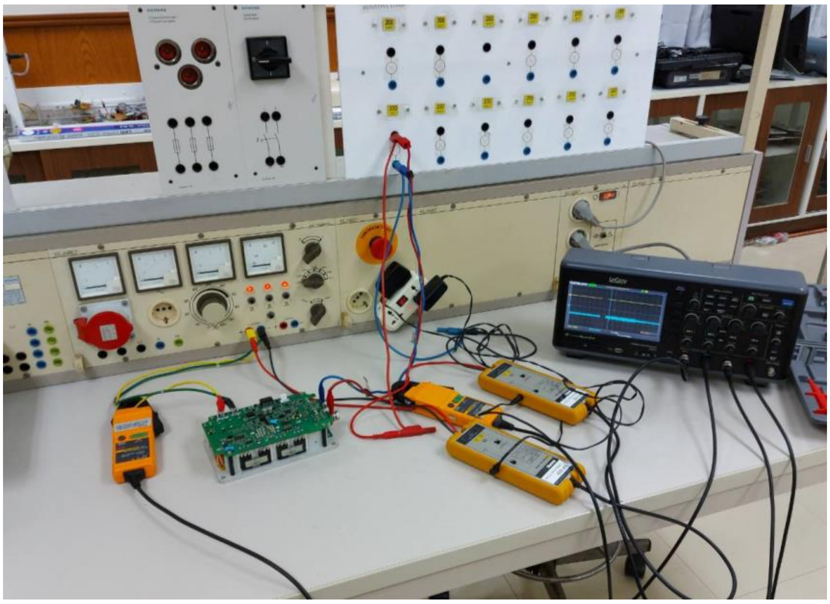


Figure 3.7.2 EV DC - DC Converter Testbench at Power Electronics Laboratory

The Vcc activation power off, there will be no communication between the host computer and converter. The auxiliary power off is the last procedure for powering off the EV DC – DC converter.

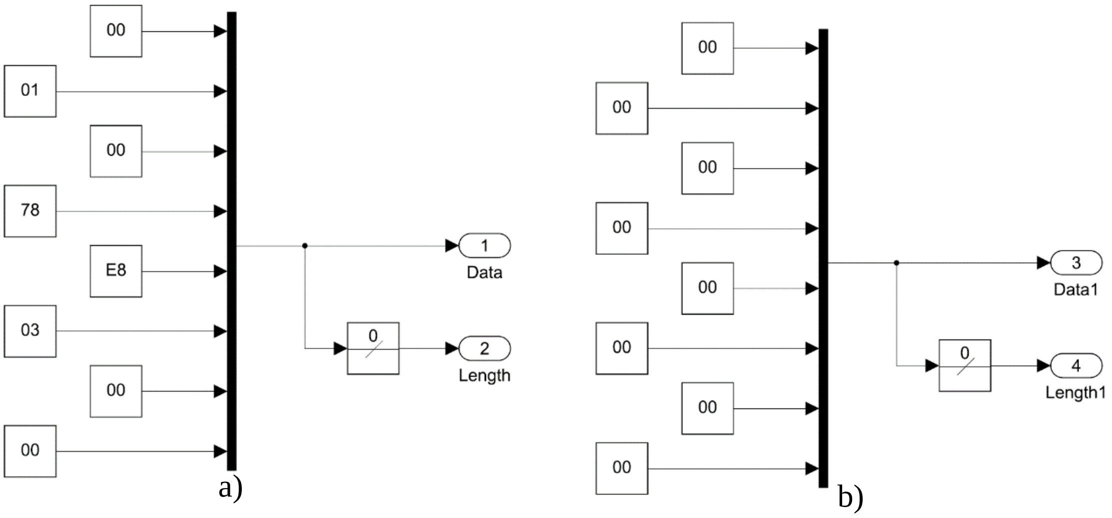


Figure 3.7.3 CAN frame a. Boot Data, b. Shutdown Data

The CAN frames transmit from the host computer to power on the converter by using the CAN protocol on VCU command. The calculation of the output voltage can be done by finding the low byte and high byte of the output value that is desired to get at the terminal. To send the off command to converter, set all the data to zero since the powering off command is 00.

4. Chapter 4

4.1. Discussion

In this project, we considered for both the company and trainee while creating the training topics. Through the project, we learnt about NXP model-based design toolbox, CAN communication, PID tuning, schematic of the NXP MPC 5744P devkit board, VCU, DC-DC converter, and DC motor.

We encountered some problems through this project, but we solved them. Since we did this project during the Covid-19 pandemic period, our project was delayed; the components we needed for the project were hard to buy and our members were not able to work well together. However, we finished the project on time by putting our best effort on it.

Furthermore, we had a problem with resetting the old programs in the NXP devkit board to upload the new program. Thus, we found the new way of uploading the new program to the controller without resetting; we use S32 software and use the elf file of our new program to upload to the controller board. The problem we faced about the DC-DC converter was that the data sheet provided by the company does not exactly match with the hardware. Thus, we carefully figured out the pins of the DC-DC converter.

We understood how the EV system uses CAN bus communication for, not just reducing the number of wires but the efficiency and reliability of using the CAN bus. We faced some problems during the implementation such as cable disconnection which confused us about our working program. From this experience, we knew that not only program error but when we implemented, we suppose to check hardware connection as well before judging the program error.

In motor control application, we used our knowledge in the control system to practice and apply the control method in the practical application. This allowed us to understand the concept and theory that was taught in class to use it for practical purposes in the engineering field. This application is very beneficial for students in their future career if they are in Electrical or Mechatronics Engineering field because most applications in automotive industrial or manufacturing industrial are motor control applications. From this training program, students will receive enough knowledge of practical application.

In EV DC - DC application, the knowledge of power electronics was applied to the commercial product that was sponsored by an electric vehicle company. Students can see the real commercial product of DC - DC converter that is used in electric vehicles. They will have the opportunity to work in the electric vehicle industry since the technology of EV is growing rapidly.

This project benefits to not only students, our trainee but also the university and the company. Assumption University of Thailand will gain the reputation since its engineering students will have higher opportunity for employment in automobile industry. Moreover, the positive reputation of the university will be increased by giving its engineering students the opportunity to be offered by some companies to work on their projects. Mine Mobility Research company can also acquire the reputation of collaborating with an international university and get the new engineers who have a proper training relating to automobile industry.

4.2. Future Work

As the continuation of the project, we plan to write a training book after we complete this project. Thus, it will assist our trainees to participate and follow easily and conveniently during the training. Moreover, after the training, they can read the book if they forget how to use Model-Based design with VCU.

In addition, we could do more possible applications and projects in the future with the DC-DC converter and VCU, MPC 5744P. The possible application that we can do is fast battery charging for EV systems. The DC-DC converter can be used as the charging driver for the battery with high input current.

As an approach to self-driving cars, an addition of an image processing system to the EV system is an option. Through image processing, the camera can read the signposts, traffic lights, objects, and moments of the livings on the road. Meanwhile, a microcontroller can read the images and decide what the car should do and send the signal to VCU to control the action of the car.

4.3. Conclusion

Due to climate change being a big issue today, the advancement of electric vehicles will be one of the priorities for many countries and the automobile industry. This will require a large number of engineers that are capable of working with EV technologies. This where our project plays an important role since it is based on a technology that plays a crucial part in the EV as a whole. By using model-based design for programming the NXP devkit- MPC5744P and the Vehicle Control Unit we learned a great deal of theoretical knowledge and implemented it on real-world applications. And by providing that knowledge in a training course to the other students at the VME will benefit greatly not only the students themselves but our community as a whole.

References

[1] Energy Absolute Company Website: https://www.energyabsolute.co.th/homepage.asp

[2] Model Base Design Toolbox (NXP): https://www.nxp.com/design/automotive-software-and-tools/model-based-design-toolbox:MC_TOOLBOX

[3] Schematic of Devkit-MPC5744P: https://www.nxp.com/downloads/en/schematics/SCH-29333-E.pdf

[4] Data Sheet of Devkit-MPC5744P: https://www.nxp.com/docs/en/data-sheet/MPC5744P.pdf

[5] Data Sheet of VCU (NXP MPC5744 Processor): https://www.ecotrons.com/files/VCU-EV2274A%20Datasheet.pdf

[6] Pulse Width Modulation: https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html,

[7] Power Electronics Devices, Circuits and Application, 4th Edition, Muhammad H. Rashid

[8] CAN Communication: https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en

[9] CAN Communication: https://www.kvaser.com/about-can/the-can-protocol/can-messages-33/

[10] CAN Communication: https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/

[11] CAN System Engineering from Theory to Practical Applications, 2nd Edition, Lawrenz, Wolfhard Lawrenz (Ed)

[12] BUSMASTER: http://rbei-etas.github.io/busmaster/

[13] "Feedback Control of Dynamic Systems", 5th Edition. Upper Saddle River, NJ, 2006, Pearson Education, Inc, G.F. Franklin et al

[14] "Modern Control Engineering", 5th Edition, Katsuhiko Ogata

[15] "Control System Engineering", 6th Edition, Norman S. Nise

[16] For Fig 1.1 and 1.2 : http://www.eng.au.edu/index.php/1st-model-based-design-training-8-9-june-2019/

[17] For Fig 1.4: https://zhuanlan.zhihu.com/p/36654874

VITA

Mr. SOTHEANITH VANNY

Electrical and Electronics Engineer
303 Moo 4,
Bang Bo, Samutprakarn 10560
+66 917532880
sotheanithvanny@gmail.com

Objective

Seeking a position as Electrical and Electronics Engineer Position which is suitable for me and could apply my knowledge and experience in.

Summary

Expect to graduate from department of electrical and electronics engineering in October, 2020 with the strong mindset, confidence, ethical of engineer and the knowledge of high voltage, power electronics, control system, design and construct the embedded system, electrical and electronics equipment's testing, power system analysis, electrical installation system design, power system protection, electric drive design system and motor control application with analog and digital. An experience working on the senior project sponsored by electric vehicle company and have been to industrial training workshop for engineers.

Profile

Name:	Sotheanith Vanny
Sex:	Male
Nationality:	Cambodian
Date of Birth:	November 9, 1998
Health:	Excellent
Place of Birth:	Battambang, Cambodia
Status:	Single

Education

- Bachelor of Engineering (BE) in Electrical and Electronics Engineering
  From 2016 - expected to graduate in October 2020
  GPA: 3.07
  Assumption University of Thailand
- Baccalaureate Education
  From 2013 – 2016
  Dewey International School, Battambang, Cambodia
- General Education
  From 2010 – 2013
  Dewey International School, Battambang, Cambodia

MIN KHANT SOE

Mobile No: 09-91912201
Email: minkhant.ppn@gmail.com
Address: InterView Abac Bangna, 189 Soi 1, Tamban Bang Bo, Amphoe Bang Bo, Chang Wat Samut Prakan, 10540

Engineering student with excellent communication and interpersonal skill with GPA of 3.82. Seeking to apply mechatronics background to get experience in Control Systems, Automation, or Industrial Robotics. Strong PC skills and experience in PLC design.

PERSONAL DETAILS

Date of Birth: 9th May, 1996 (24Yrs)
Marital Status: Single
Passport No: MB695370
Nationality: Myanmar

LANGUAGES

English: Intermediate Level

SKILLS & ABILITIES

AutoCAD 2D, CX-Programmer, CX-Designer, NB-Designer, MATLAB and Simulink, Ladder programming, C programming

WORKING EXPERIENCE

Student Affair in Assumption University Myanmar Student Community (AUMSC) (May 2017 - April 2018)

- Help Myanmar students in the academic process
- Book the rooms for Myanmar events and football pitch for friendly matches.
- Help the AUMSC team in creating Myanmar cultural events such as homage paying ceremony in Thadingyut Festival, Thingyan Festival, etc.

Assistant Teacher in programmable logic controller (PLC) lab (Jan 2020 - Sep 2020)

- Assist the students in lab
- Was assigned to teach 2 PLC tutorial classes

EDUCATION

Degree	University	From	To	Grade
Bachelor of Engineering (Mechatronics) Final Year	Assumption University, Bangkok, Thailand	2017	Current	CGPA 3.82
		❖ Expected Graduation Date: 21st Oct 2020 ❖ Awarded Scholarship for 2nd Year (Sem 2) & 3rd Year (Sem 1)		
Bachelor of Arts (English Specialization)	University of Distance Education Yangon, Myanmar	2013	2016	CGPA 3.1

OTHER QUALIFICATION

Qualification	Awarded Bodies	Awarded Date
Advanced Technician Diploma in Engineering (Mechanical & Electrical Engineering Theory - Control Systems)	City & Guilds UK	16th Sep, 2015
Level 3 IVQ Technician Diploma in Engineering (Mechanical & Electrical Engineering - Mechatronics)	City & Guilds UK	21st Sep, 2014
Technician Diploma in Engineering (Mechanical & Electrical Engineering Theory - Mechatronics)	City & Guilds UK	11th Jul, 2014
Certificate of Participation in “A World without Engineers” Workshop	UCSI University	3rd Aug, 2013

Ivan Lenskikh

Email: u6015222@au.edu

Education

Assumption University of Thailand

Expected to graduate in March 2021 with a Bachelor of Electrical Engineering

Current GPA 3.87, completed 121 credits

Burapa English-Program School of Thailand, Pattaya

Graduated in 2017