# EE302 Assignment

Car Airbag System

Krystian Stratynski (23458994) & Jonathan Nyasulu (23477396)

Maynooth University

National University of Ireland Maynooth

# Contents

# Part 1

## Details

**Fosc**: 4Mhz

**POT1**: potentiometer representing the impact level (RA0/AN0) | Input

**Buzzer:** impact buzzer as an output (RC2) |Output

**LED1**: impact alarm LED as an output (RC0) |Output

**LCD**: 2x16 alphanumeric LCD display (RD0-RD7) | Output

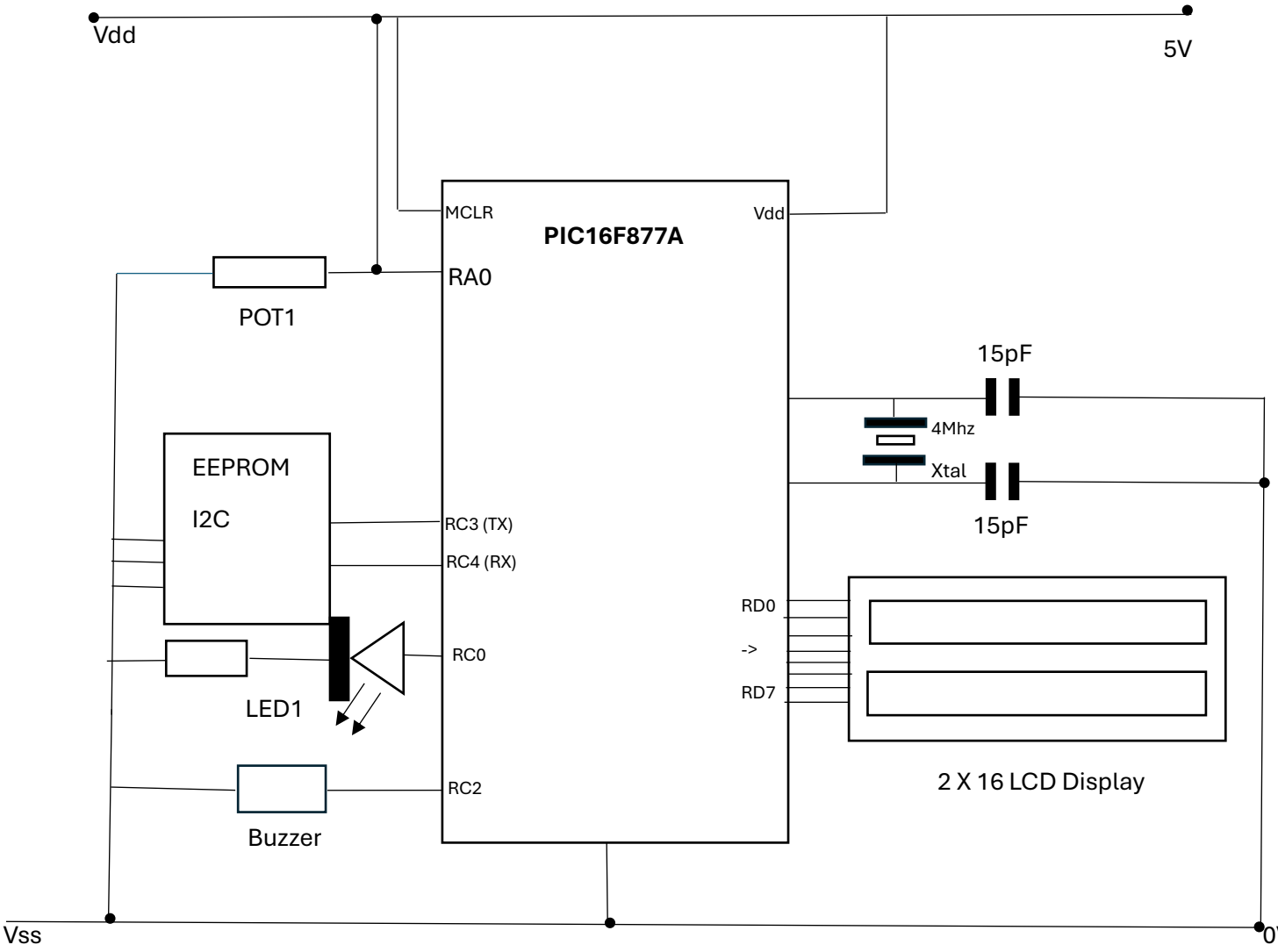
**SCL**: $I^2C$ serial clock (RC3) | Set as Input

**SDA**: $I^2C$ serial clock (RC4) | Set as Input

**DataOut**: Serial Data Output (RC3) (TX) | Set as Input

**DataIn:** Serial Data Input (RC4) (RX) | Set as Input

## Diagram



# Part 2

## Brief Overview

Our Airbag/Impact detection system comprises of a: PIC16F877A microcontroller Potentiometer (Analog Sensor), Putty, Buzzer and an LED, along with UART communication and EEPROM storage.

When the potentiometer is turned to quickly it replicates the same effect from an impact force would against an object. This is as there's a sharp change in voltage. The ADC readings from the Potentiometer are then sent onto the Putty / UART in real time. This then activates the Buzzer and the LED, further showcasing that an impact force has been detected. The Putty then logs these readings to the EEPROM

After some research and investigation, we found out that, and airbag takes a total of around 30-40ms to fully deploy after impact. 10-15ms is taken for the detection then another 15-20ms for the deployment and inflation of the airbag.

In relation to our project, we hope and aim to achieve accurate enough readings from when impact it detected to when the airbag is deployed. The Buzzer and the LED will be used to represent the Airbag deploying.

## Timers & Interrupts

In our project we used Timer1, which is a 16-bit hardware timer inside the PIC16F877A. Being 16-bit means it can count from 0 to 65,535 before overflowing compared to the 8 bit from timer0 and timer1.

Timer1 was configured to generate a periodic overflow interrupt.
When Timer1 reaches its maximum count and overflows, it sets the TMR1IF flag. With interrupts enabled, this triggers the Timer1 Interrupt Service Routine, where we start our ADC impact-detection reading cycle.

We configured Timer1 with a 1:4 Prescaler and preloaded it with the value 53,036, causing it to overflow every 50 Ms. This value was derived by getting the number of counts.

65,535 (max amount before overflow) – 12,500us (50,000/4 [prescaler value]) = 53,036.

So loading TMR1 = 53,036 means it will take 12,500 ticks to overflow, which equals 50 Ms.

This is the bit config we got based on the 1:4 pre-scaler

T1CON = 0b00100001; from rightmost highlighted digit to leftmost:
Timer 1 on, internal clock (fosc/4), Prescaler 1:4