Practical 1 Lab Book Git Test

# Team members:

Armandt Fourie – 17028002

Franco van Tonder – 1707594

Henko Borstlap – 15010563

Albertus Richter – 17020353

# Mission Critical Design Specifications:

## General specifications:

1. The MARV should use either a PIC18F45K22 or a PIC18F45K20. We will use the former.
2. The MARV should only use the motors that were given, we therefore have four to our disposal.
3. The MARV should not exceed the dimensions 10cm x 18cm x 12cm (WxLxH) except for the “chimney”.
4. The MARV should survive a drop from a height of 50cm.
5. The MARV should use either one or two rechargeable 9V batteries.

## Technical requirements:

1. Must be able to reliably distinguish between the colour white, and any of the following: green, blue, red and black in order to determine whether the correct line is being followed. It must do this with **95% accuracy**.
2. The MARV must complete the race track in less than three minutes [cite prac guidelines]. On a 5m track, this means it should have a minimum average speed of **0.03m/s**.
3. The MARV must be able to detect when a line curves and turn in the correct direction within **100ms**.
4. The MARV must continue to function regardless of ambient light – the MARV must be able to follow a line even if lighting conditions are changed during runtime.
5. The MARV should stop within **10cm** of the end of the track automatically.
6. The MARV should be able to detect a line beneath it within **one second**.
7. It should take less than **one minute** to calibrate the MARV.

## Hardware concept design

**Subsystems:**

Power Supply Subsystem:

This subsystem must provide power to all the components of the MARV. It will receive power from either one 9V battery and must convert this to a 5V DC output. The PIC will be powered by this 5V supply, for example.

Testing:

1. Using a multimeter or an oscilloscope, it should be confirmed that this subsystem outputs a 5V DC signal.

Sensor Subsystem:

This subsystem will accept receive power from the power supply subsystem. This means that this subsystem will be powered by a 5V DC power supply. This subsystem will be used to detect the colour/presence of a line beneath the MARV using an array of phototransistors with accompanying LEDs. This subsystem therefore accepts DC voltage and outputs an amplified signal that can be processed to determine the colour detected by each sensor.

Testing:

1. The LEDs of this subsystem must light up.
2. This subsystem must output a voltage depending on the colour track beneath each phototransistor. The voltage must therefore change if the sensor is exposed to a different colour.

Start/Stop Button:

The start/stop button is responsible for starting and stopping the MARV. While in race mode, the MARV will initially wait for instructions given through the serial communications subsystem. If the start/stop button is pressed while the MARV is in static race mode, the MARV must transition into the moving race mode, where it rides along the lines of the racetrack. If the button is pressed while the MARV is in moving race mode, the MARV will stop and transition into static race mode.

Testing:

1. If the button is pressed while the MARV is in static race mode, the MARV must start moving.
2. If the button is pressed while the MARV is in moving race mode, the MARV must stop moving.

Serial Communications Subsystem:

This system provides the terminal-based interface between the team and the MARV. This is used to transition between states and to set the welcome message. This is facilitated by a USB-to-serial bridge. The input to the bridge is connected to a PC, while the output is connected to the PIC microcontroller. This system must receive strings of arbitrary size and transmit them to the PIC. It must also be able to transmit string of arbitrary length from the PIC to the PC. Part of this subsystem will be implemented in firmware: depending on the input received, the MARV’s firmware must change between states that have their own test criteria. To test whether the system is functional, two LEDs will be used. One will be lit up to indicate that serial communication *to* the PIC was successful and the other will be lit up to indicate that serial communication ­*from* the PIC was successful.

Testing:

1. After a message is transmitted from the PC connected to the USB-to-serial bridge, the *­serial receive* LED lights up.
2. After a message is received from the USB-to-serial bridge on the PC, the *serial send* LED lights up.

Data Storage Subsystem:

This subsystem will be used to store string to an EEPROM module. This module will be powered by the Power Supply Subsystem. This system is also mainly implemented using firmware. The functionality of this subsystem will be confirmed using two LEDs much like the Serial Communications subsystem. After a string is saved to EEPROM, one LED will be lit up and the other LED will be lit after a string is received from EEPROM and transmitted via the Serial Communications subsystem to the team member on a PC. The team member will be able to read the string (which is decided by the team) and can check the LED to confirm that a string was transmitted.

Testing:

1. After a string is transmitted from the EEPROM connected to the PIC, the *I2C receive* LED lights up.
2. After a string is received from the PIC to the Serial Communications subsystem, the *I2C send* LED lights up.
3. Following user prompts, the string (welcome message) is received from the PIC and displayed in the terminal.

Motor Subsystem:

This subsystem is used to propel the MARV along the track. This subsystem receives its power from the Power Supply subsystem and will be controlled by the PIC according to the firmware. This system entails four electric motors that will be controlled in groups of two, based on the side of the MARV that they’re on. In order to move forward, the MARV will make all motors turn in the positive direction. In order to turn right, the left motors must turn in a positive direction while the right motors turn in a negative direction. The opposite is true for turning left.

## Assignment of Minimum Requirements:

Armandt – sensor calibration subsystem

Bertie – Serial communications (receive string through terminal, return sentences)

Henko & Franco – Firmware state machine design (therefore large chunks of the debugging system), dump register to output port, read and write to EEPROM

## Notes on the Calibration Subsystem:

Outputs from this subsystem: ̅

* Display characters on SSD.
* Flash LED after each calibration.

Inputs:

* “CAL” message from PC.

As specified in the practical guideline [cite prac guideline], the system must repeat the following steps five time (once for each colour):

1. Display colour character on SSD (B, R, G, |\_|, | ̅ |).
2. Delay for 3s (this replaces the actual “calibration”. In this practical we will simply take a signal from the ADC and store it in a register as well as implementing a 3s delay).
3. Flash an LED corresponding to the colour.