CG1111 Engineering Principles and Practice I

The A-maze-ing Race Project

Project Group: Section A - Group 07

Group Members

Lucas Foo Soo Quan (A0182390E)

Mohamed Riyas (A0194608W)

Mohideen Imran Khan (A0181321U)

Shriya Saxena (A0194079R)

Parvathi Ranjith Menon (A0194448R)

Introduction

This report documents the various design decisions made by our team for the *A-maze-ing Race*.

The outline of the report is as follows:

1. Overall algorithm
2. Robot alignment
3. Challenge detection
4. Colour detection
5. Sound detection

Additionally, we include details on labour division and significant difficulties faced during the preparation of the mBot.

Overall Algorithm

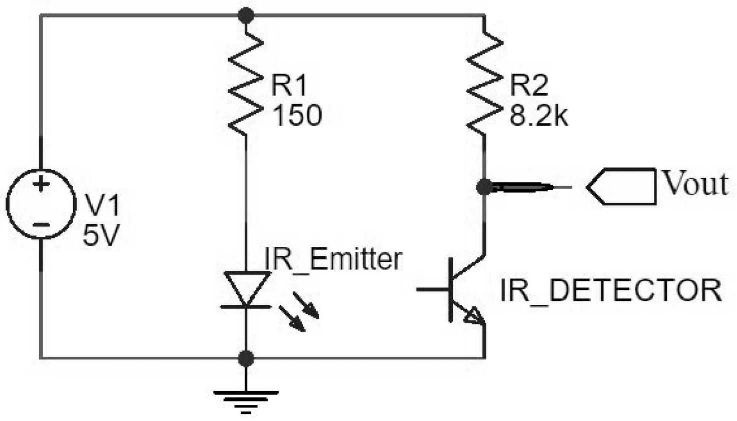
Our overall algorithm is summarised in the pseudocode that follows.

```
1. If there is a challenge:
    a. Check the colour of the board above.
    b. If the colour is not black:
        i. Perform colour challenge.
    c. Else:
        i. Check if there is an audio input.
        ii. If there is an audio input, perform the audio challenge.
        iii. Else, play victory tune and terminate the program.
2. Else:
    a. Align the robot and repeat from step #1.
```

We made the decision to check for the presence of colours before proceeding to check the audio input because our colour detection algorithm proved to be more reliable than the audio detection circuit.

Robot Alignment

Two breadboards with the circuit diagram shown on the right were mounted on the left and right sides of the mBot.



It follows from the circuit arrangement that V_{out} will drop when an obstacle (the maze wall) nears the breadboard and vice versa. V_{out} was connected to an analog pin on the mCore via the RJ25 adapter and output from the ADC was used without conversion.

Since the two circuits on the left and right sides of the mBot tend to produce different output voltages even when the distance to the maze walls is almost the same, we made the decision to calibrate the sensors whenever the mBot was switched on. We asked the robot operator to place the mBot in the middle of two maze walls, and ran the calibration algorithm as detailed:

1. Measure the output from the left IR circuit. Label this measurement D_{left}.
2. Measure the output from the right IR circuit. Label this measurement D_{right}.

Our alignment algorithm is thus:

1. If the input from the left IR sensor is less than (0.9 x D_{left}), steer the robot towards the right.
2. Else, if the input from the right IR sensor is less than (0.9 x D_{right}), steer the robot towards the left.
3. Else, move forward.

Note that 0.9 is a parameter obtained by experimentation. For our group, 0.9 allowed our robot to move smoothly without jerking while still preventing bumps.

Turns (left, right, left and left, right and right, U-turn) were programmed through timings. We measured the time required for the robot to make the appropriate turn and entered that measured timing into our program. This method has a drawback that we discuss in the "Difficulties Encountered" section.

Challenge Detection

The line sensor mounted at the front of the robot was used to check for the presence of challenges (black lines). When both the light sensors in the sensor unit detect black, the robot stops and performs the challenges in the order described in the "Overall Algorithm" section.

As such, our algorithm for challenge detection is as described:

```
1. Function isChallenge():
    a. Return true if both the light sensors in the sensor unit sense black.
    b. Return false if otherwise.
```

Colour Detection

The RGB LEDs and Light-Dependent Resistor (LDR) on the mCore were utilised for the colour challenge.

To make measurements, the LEDs were set to red, green and blue in succession with a 200 milliseconds delay to allow the LDR output to stabilise. For each colour, 5 measurements were made with a 10-millisecond interval and the average was taken.

For calibration, the RGB values for black and white colours were noted and from these measurements, a derived RGB value, named range, was calculated. The RGB values for black and white were manually obtained and typed into the program on a regular basis, with the final assignment done on the eve of the race.

When the colour of a sample needs to be determined, a raw measurement would be made and the RGB value would be calculated as follows:

1. For each colour in {Red, Green, Blue}:
    a. Colour value = (raw reading for the colour - black reading for the colour) / range for the colour x 255

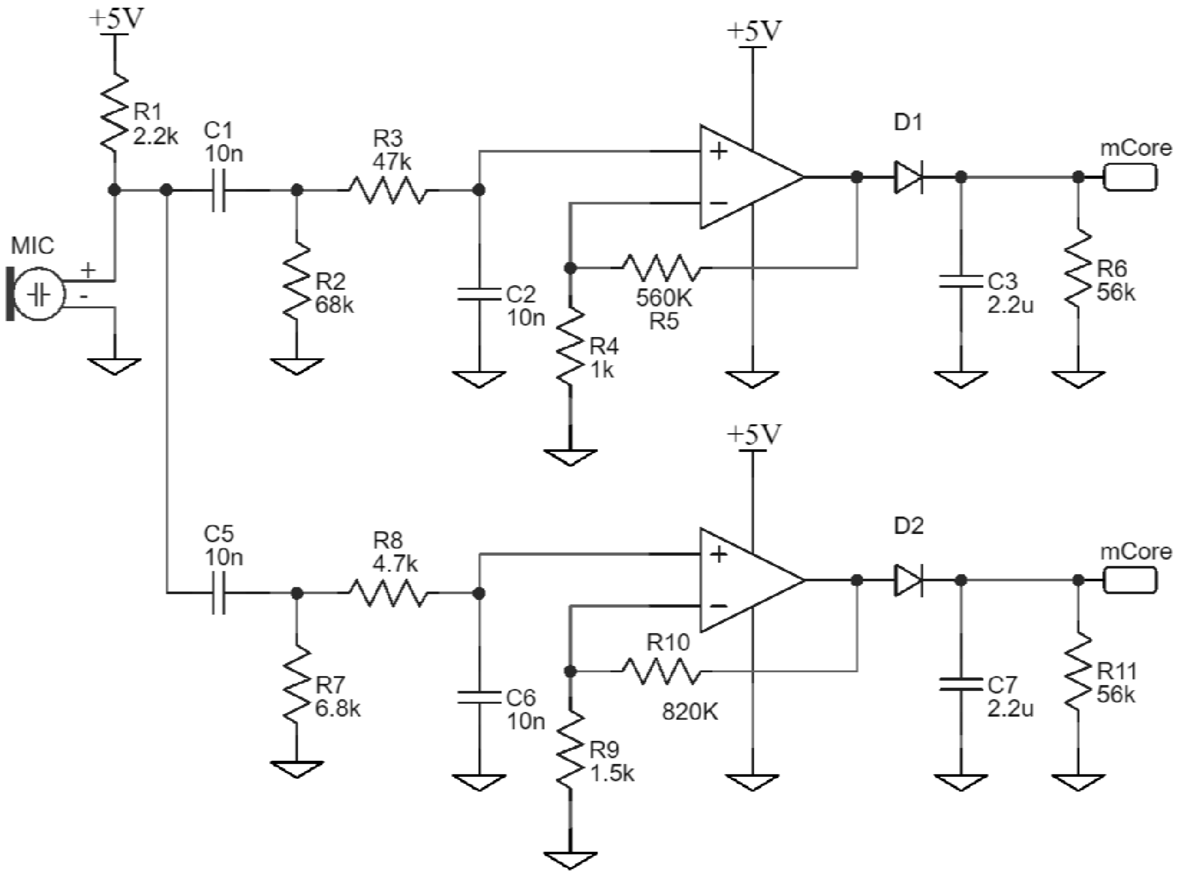
Additionally, we defined the RGB values of the possible colours (black, white, green, blue and red) in an array. These values were obtained by manually making repeated measurements of the colour samples (through Arduino Serial out) and taking the average.

Let us now refer to the RGB value of any colour to be a set of coordinates. To rephrase the previous paragraph, we defined points in 3D space for each of the possible colour with the x-,y- and z-axis corresponding to the red, green and blue values respectively.

Now, when a colour needs to be determined, we treat it as a new point in our 3D space and find the nearest predefined point. Here, nearest refers to the Pythagorean distance. The colour corresponding to the nearest predefined point is then taken to be the colour of the sample. The action associated with each colour is then performed unless the colour is black. For the black samples, the audio challenge was invoked to check for the presence of audio signals.

Sound Detection

The following circuit was utilised for the sound challenge:



The circuit contains the following components:

1. A band-pass filter to detect frequencies between 234Hz and 331Hz.
2. A band-pass filter to detect frequencies between 2340Hz and 3316Hz.
3. An op-amp for each band-pass filter to amplify the filtered signals.
4. A peak detector for each op-amp to convert the rectified sinusoidal output from the op-amp to a stable DC (almost) value.

The outputs from the two peak detectors, P_{300} and P_{3000}, were connected to analog pins on the mCore via an RJ25 adapter. Like the inputs from the IR sensors, the input from the ADC was used directly.

The algorithm is thus:

1. If P_{300} is greater than the threshold value for 300Hz and P_{3000} is greater than the threshold value for 3000Hz, perform a U-turn.
2. Else, if P_{300} is greater than the threshold value for 300Hz, make a left turn.

3. Else, if P_{3000} is greater than the threshold value for 3000Hz, make a right turn.
4. Else, play victory tune and terminate the program. Note that the audio challenge is invoked only when a black sample is detected and as such, this is an appropriate action.

The thresholds for P_{300} and P_{3000} were obtained experimentally. We had to adjust the thresholds when the volume of the speaker was changed.

Difficulties Encountered

As mentioned earlier, the turns were timed. For instance, the robot spun in the clockwise direction for 1100ms to turn right. However, the speed of the turn depended on the battery charge. The robot turned more than 90 degrees when the battery was fully charged and less than 90 degrees when the battery level was low.

To overcome this problem, all timings were made when the battery was almost fully charged, and the race was performed with fully charged batteries. An encoder would have allowed for more reliable turns.

Secondly, the audio circuit proved challenging to build. We were initially confounded by the lack of a sinusoidal waveform on the oscilloscope when playing a 300Hz sound. However, we soon realised with the help of Dr Sangit that the signal was present but was clouded by noise from the surroundings, the instruments and the circuit itself. Thereafter, we used the FFT function of the oscilloscope to detect the presence of the signal and used it extensively to build and tune the circuit to the desired state.

Next, finding a suitable threshold for IR sensors was difficult. The Arduino's ADC returned significantly different values for the two IR sensors even though the mBot was placed in the middle of the maze walls. Therefore, the decision was made to calibrate the IR sensors every time the mBot is switched on. This greatly reduced the chances of the robot bumping into the maze walls.

Division of Labour

Lucas -- Robot alignment and motion

Riyas, Parvathi, Shriya -- Audio and IR circuit

Mohideen -- Colour detection

Acknowledgement

We would like to thank Dr Sangit Sasidhar, Dr Colin Tan and teaching assistant Vincent for making this project possible and enjoyable.