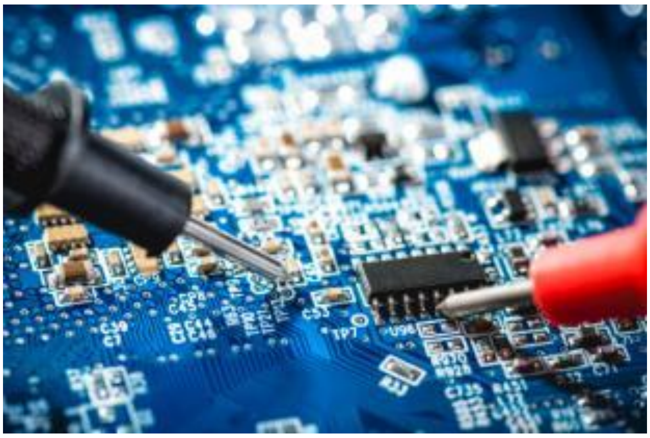


EEBUG DESIGN PROJECT

STAGE 2 REPORT

ABSTRACT

EEBUG Group Design Project was awarded to the 1st Year students of Department of Electrical and Electronic Engineering. This report documents how the work progressed during the Stage 2 of the Project with a focus on eebug enhancements.

Mariam Sarfraz;Haosheng Qian;Yanni She

Department of Electrical and Electronic Engineering, Imperial College of Science Technology and Medicine

Contents

1. Project Management

1.1 Project Overview

The main objective of the EEBUG project is to design and build an improved EEBUG.

This enhanced bug is supposed to:

- Follow a marked track, of width approximately 6mm, in fading greyscale on a white background
- Leave a mark, using a marker or pen, to show the path it has followed
- Continue moving in a straight line for 10cm after the track has ended
- Draw a spiral with decreasing radius, completing at least one full turn before coming to a halt

The budget of each bug is £8 and this amount cannot be exceeded. We are required to not use more than two sensors and a maximum of 4 AA batteries can be used. Once the bug is turned on, we cannot intervene and it should follow the path stated above on its own.

To build such an enhanced bug, we were required to analyze the original bugs we built in the first term laboratory experiment. The internal circuitry of the bugs was to be studied in detail, comprehending the behavior of all electronic components used. Once the original bug was fully analyzed, we were required to compare the bugs and select the components that fit best with the project objective. The positioning of the sensors was required to be selected. The enhanced circuit was to be built and analyzed. The microcontroller was to be carefully selected and programmed.

1.2 Project Progress

1.2.1 Overview of the work done Last Term and stage 1 Report

We started by analyzing each of our bugs in depth and comparing how the various different components operate.

After comparing how the motor speed varies with the light intensity for each bug, the pros and cons of the bugs were compared and we decided on using the combination of NPN transistor and LDR because it gives a bug which is sensitive enough to detect the path and is easy to control. Both the practical and simulated results were taken into consideration.

In our first report, we had decided to divide the route of the bug into three different parts:

1. The bug follows the track. The route is detected by a two sensors. Each sensor contains an LDR, to detect the light intensity, and an LED, to emit the light.
2. When the track ends, the bug moves straight for 10 cm. This is because the velocity of the bug depends on the constant surface.
3. The microcontroller used will have a wheel revolution recorder. After a particular number of revolutions had been made, the motor velocities will decrease at different rates and finally, the bug will come to a halt.

Full details of this were provided in the first report.

1.2.2 Details of the Progress made since last Management Report

This term, progress has been made by building the bug with the components and sensors shortlisted last term. Our team then designed, built and tested the final circuit. The microcontroller was wisely chosen and bought (details are in section 2) followed by the programming of the microcontroller. Finally, the EEBUG was tested and analyzed in depth.

1.3 Project Planning

1.3.1 Communication Methods

Meetings were held on a regular basis to ensure that progress was being made at a steady pace. The meetings were kept on every Wednesday from 13.00 to 15.00 and every Saturday from 14.00 to 17.00. A WhatsApp group and a Facebook Group were created to eliminate communication gaps between members. Time, date and venue of meetings was decided through these groups as well. A Dropbox folder was also created so that members could upload their respective work in the folder and hold all documents in one place for easy accessibility for all team members. A folder was created to hold details of each meeting and its minutes and was kept in a EEE locker so that all members could have access to it at all times. If a member missed out on more than two meetings for an invalid reason or was unable to perform his/her required task in the given time, a red flag was issued. If there was a conflict between members, voting was done to settle the issue.

1.3.2 Aims and Milestones

A work plan was developed that outlined the decision of work between the three team members and specified the timelines and deliverables for each. Progress was monitored against the defined milestones.

<u>FIGURE 1</u>

PROJECT GANTT CHART

			2015												2016																			
S.NO	ACTIVITIES	LEAD RESPONSIBILITY	OCT				NOV				DEC				JAN				FEB				MAR				APR				MAY			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Review of the background reading materials					X	X																											
2	Defining roles							X																										
3	Defining the work plan and deliverables for Autumn Term							X																										
4	Defining the respective roles and responsibilities for Autumn Term							X	X																									
5	Analysis of original bugs							X	X																									
6	Comparison of bugs									X																								
7	Work on design enhancement									X	X																							
8	Client meeting and its review											X																						
9	Working on the stage 1 report								X	X	X	X																						
10	Submission of stage 1 report												X																					
11	Circuit Design															X	X	X																
12	Circuit Implementation																	X	X															
13	Microcontroller Programming																			X	X													
14	Microcontroller Implementation																					X												
15	Testing and Optimization																					X	X											
16	Testing and Sensor Calibration																						X	X										
17	Working on the stage 2 report																			X	X	X	X	X										
18	Submission of stage 2 report																								X									
19	Switch Design																								X	X	X							
20	Penholder Design																											X	X					
21	Superbug functions																												X	X	X			
22	Preparation for the Design Project Viva																														X	X		

Table 1: Activities and Deliverables

Timeline	Activities	Deliverable
Week 3	• Discussion on work in hand and how it should be distributed amongst the team members • Allotment of respective roles • Developing the outline of the report • Developing the work plan to be followed	• Team structure and responsibilities defined • Second stage report format defined • Gantt developed
Week 4	• Designing the circuit • Circuit implementation	• Final internal circuitry of the bug built
Week 5	• Selecting which microcontroller to buy and use	• Microcontroller selected
Week 6	• Programming the microcontroller i.e. writing the microcontroller algorithm	• Bug working as per instructions programmed through the microcontroller
Week 7	• Testing for optimization • Troubleshooting	• Enhanced bug
Week 8	• Initializing work on the stage 2 report	• Report Structure defined
Week 9	• Discussing and agreeing upon the future plans • Deciding on the superbug function • Final testing	• Future work proposal in place
Week 10	• Compilation of the final draft of the stage 2 report	• Stage 2 report drafted and submitted

1.3.3 Roles and Responsibilities

In order to give the team a firm management structure, the workload was divided equally between the three team members

Table 2: Roles and Responsibilities

Team Member's Name	Team Member's Roles and Responsibilities
Mariam Sarfraz	▪ To lead the group i.e. appointing and managing all tasks and making sure that the work plan is followed effectively ▪ To arrange the date and time for meetings ▪ To write an overview of all the meetings including their dates and minutes ▪ To draft the first two sections of the report, comprising of the Project Management (20%); the High level design (20%);and ▪ To draft the last section of the report (10%) ▪ To compile and edit the final draft of the report
Haosheng Qian	▪ To compile and upload the final draft of the report ▪ To efficiently compile all the documents related to the work on the EE bug ▪ To draft the third section of the report together with Yann i.e. the Detailed Design (35%); and ▪ To draft the fourth section i.e. EEBUG enhancements and testing (15%)
Yann She	▪ To manage and record the entire cost of the project effectively ▪ To ensure that all simulated and experimental graphs and data are correct ▪ To manage the team's funds in a prudent and efficient manner ▪ To draft the third section of the report together with Haosheng i.e. the Detailed Design (35%)

1.4 Proposals for future work

In week 9 of the Spring term, our team discussed and planned the future activities that will be implemented after the Spring term.

These included:

- Designing of the penholder for the EEBUG so that the path followed by the bug can be tracked;
- Designing and implementation of the Superbug function;
- Final testing and optimization of the EEBUG; and
- Preparation for the individual project viva

2. High Level Design

2.1 Description of proposed solutions

Detecting Stage

I. Using an LDR without an LED
Normal light would be used as a source. This would save money and will offer a more economical solution.

II. Using an LDR with an LED
This would ensure a stable light source.

III. Using an IR sensor
This would block all the noise coming from the surroundings. However, during the digital-to-analogue conversion in the microcontroller, we can block the noise anyways. Hence, this may not be required.

'Following the Track' Stage

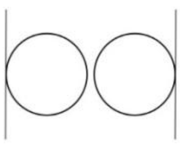
I. Setting the sensors on the white surface as reference

Figure 2

As shown in Figure 1, the set up works to follow the path. However, it cannot detect when the path has completely faded and thus does not know when to switch to its next function.

II. Setting the sensors on the black surface as reference

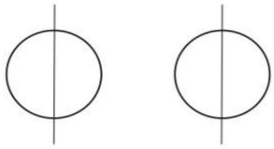
Figure 3



The diameter of the path is approximately 6mm. As it is too thin, the position displayed in Figure 2 cannot be used as a reference point. The line is not thick enough for us to execute this set up.

III. <u>Setting the sensors in between the white and black surface as reference</u>

<u>Figure 4</u>



The set up displayed in Figure 3 is the reference position that we obtained from our Client Meeting Feedback. However, this could also not be used as the path was too short. The system is not effective for detecting the exact middle of the line.

Microcontroller Selection Stage

I. <u>PIC 12F629</u>
This was our first microcontroller selected. The drawback was that it did not have an analogue-to-digital convertor.

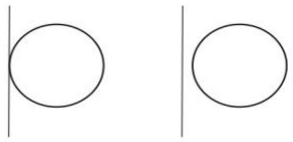
II. <u>PIC 12F1840</u>
This was our second selection. Although it had an analogue-to-digital convertor, but it had only one PWM (Pulse Width Modulation).

III. <u>ATTINY85-20PU</u>
This microcontroller had two PWMs.

2.2 Final Selection

'Following the Track' Selection

<u>Figure 5</u>



The set up shown in Figure 4 was our final selection. It not only follows the path but can also detect the end of the path. The width of the path is thick enough for us to adjust one sensor on it completely.

Microcontroller Selection

ATTINY85-20PU:
This was the best choice as it had 2 PWMs. Other than that, we could use the language ARDUINO, which is very similar to C++. Hence, the program was much easier to write and compile.

Circuit Components Selection

Q. Should we use a transistor or not?

Ans. As the output current of the microcontroller cannot exceed the maximum microcontroller current limit i.e. 40 mA[1], a transistor in series with a motor can be used to work like a switch. The base of the BJT transistor is connected to a resistor which limits the output current of the microcontroller.

Q. Should we include a capacitor or not?

Ans. Every time the motor runs, a back E.M.F is produced. This will affect the whole circuit, making the circuit unstable. Hence, we should add two capacitors across the power supply to enhance stability of the circuit.

2.3 Total cost of the enhanced bug

1. Supplier: Onecall
   Reference: 9758992
   Description: MICROCHIP PIC12F629-I/P 8 Bit Microcontroller, Flash, PIC12F6xx, 20 MHz, 1.75 KB, 64 Byte
   Quantity: 1
   Price: £ 0.93
   NOT USED. Does not have the essential features i.e. no digital analogue convertor

2. Supplier: EEDStore
   Description: ELECTROLYTIC RADIAL 470nF 35V
   Quantity: 2
   Price: £ 0.20
   The motor generates a back EMF in the circuit, which is proportional to the motor rotational speed. If the speed changes, the EMP will change leading to the change of current. If the speed change is unpredictable, you will have a noise in the current. The current change in the circuit will cause change in the voltage of power supply, so the capacitor is used to reduce this change.

[1] Obtained from the User Booklet of ATTINY85-20PU

3. Supplier: EEDStore
   Description: Resistors
   Quantity: 6
   Price: £ 0.24

4. Supplier: EEDStore
   Reference: VB0030
   Description: STRIPBOARD SMALL 25mm X 64mm
   Quantity: 1
   Price: £ 0.3
   To place the LEDs

5. Supplier: RS
   Reference: 696-2327
   Description: Atmel ATTINY85-20PU, 8bit AVR Microcontroller, 20MHz, 512 B, 8 kB Flash, 8-Pin PDIP
   Quantity: 1
   Price: £ 1.08
   The microcontroller works like the brain of our circuits. Taking inputs, doing calculations and giving outputs.

6. Supplier: EEDStore
   Reference: SD0030
   Description: DIODE ZENER 1.3W BZX85 4.7V
   Quantity: 1
   Price: £ 0.15
   NOT USED. This voltage is not high enough to meet the requirement.

7. Supplier: EEDStore
   Reference: SW0150
   Description: MINIATURE 240V SPDT VERTICAL PCB SLIDE
   Quantity: 1
   Price: £ 0.78
   Use as a switch

8. Supplier: RS
   Reference: 625-6451
   Description: ON Semiconductor 1N5338BRLG Zener Diode, 5.1V 5% 5 W through Hole 2-Pin DO-15
   Quantity: 5
   Price: £1.1
   The battery may die when we use it, so the voltage of our power supply is not constant. As we use Vcc as our reference voltage, this may cause reference problem. Therefore, we use a zener diode.

9. Supplier: Onecall

Reference: 1497991
Description: OPTEK TECHNOLOGY OVLAW4CB7 LED, T-1 (3mm), White, 6200 mcd, 45 °, 20 mA, 3.4 V
Quantity: 2
Price: £ 1.36
Light source

10. Supplier: Onecall
 Reference: 1971858
 Description: MICROCHIP PIC12F1840-I/P 8 Bit Microcontroller, Flash, PIC12F18xx, 32 MHz, 7 KB, 256 Byte, 8, DIP
 Quantity:1
 Price: £ 1.01
 NOT USED. It only has one PWM, but we need at least two PWM to control two motors.

Total Cost: £ 7.15
Predict Cost: £ 4.18

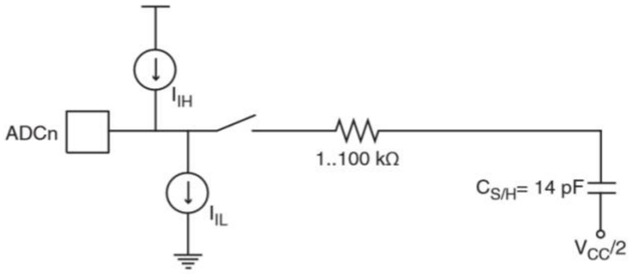
3. Detail design

3.1 High level description of enhancement realization implementation

According to the discussion on the first stage, two pairs of LEDs and LDRs will be used as the sensors of the bug. A resistor used on each side (left and right) works as a potential divider for sharing a voltage with the LDR. The voltage across that resistor will be taken as the input of the microcontroller. The input voltage will be used to refer to the Vcc and GND. In this case, the mode used for Analog-to-Digital Conversion is single-end channel.

"The input analog voltage will be used to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete."[2] *Details attached in Appendix on Input Circuitry*.

Figure 6
Input circuitry



The output of the 10 Bit-ADC will be a value passed by this equation:

$$ADC = \frac{\text{Vin}}{\text{Vref}} \times 1024$$

In this case, the output of ADC will be within the range of 0 to 1023, which is compared to the Vcc value. The output of ADC at each time can be recorded and used for checking, comparing the digital value of the two inputs (left and right), by simply doing the subtraction:

$$Diff = Vlin\ (input\ voltage\ of\ left\ LDR\ potential\ divider) \\ - Vrin(input\ voltage\ of\ right\ LDR\ potential\ divider)$$

The sign of Diff can be used to find whether the left or the right sensor has reached the black line. If the sign is negative, the velocity of the left motor will decrease and if the sign is positive, the velocity of the right motor will decrease. When the LDR reaches the black line, the black line will absorb more light than the white part. The resistance of the LDR increases. The Vlin or Vrin will decrease as the resistor in series with the LDR is sharing the voltage with the LDR. A reduction in Vlm (the output voltage of the left motor) or Vrm (the output voltage of the right motor) will cause the speed of the respective wheel to reduce. This will cause the bug to adjust its position by turning the respective wheel until the side on the white path reaches the black line and the cycle repeats again. The two LDRs will be positioned a little bit closer to achieve two purposes: to get a smooth movement of the bug and to make at least one of the LDRs detect the black line.

[2] https://learn.sparkfun.com/tutorials/analog-to-digital-conversion#res Analog to Digital Conversion

For the second stage, when the Attiny85 finds the Diff to be approximately zero, the function will be switched to give a constant velocity for a specific time that will allow the bug to go straight forward for 10cm. At this stage, Vlm = Vrm = Vmax

The output voltage of both motors will be equal and maximum.

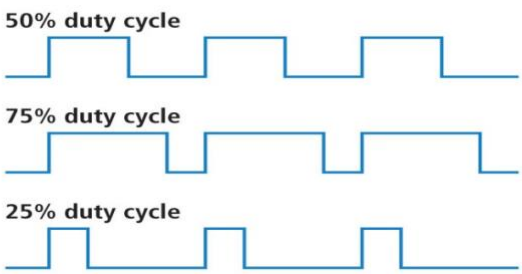
Lastly, the function will be switched for the final part. The initial value for Vlm will be assigned a higher value than the one assigned for Vrm. Both of the output voltages will decrease at the same rate until they become zero. This causes the bug to create a spiral line. The bug then finally comes to a halt. In the case above, we need an analog output to achieve the change of the speed. The Attiny85 provides a mode called PWM (Pulse-Width-Modulation) to do this task.

"Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width."[3]

If you repeat this on-off pattern fast enough with the motor, it will recognize a value between 0 to 5 Volts instead of 5 Volts. The equation of this behavior can be:

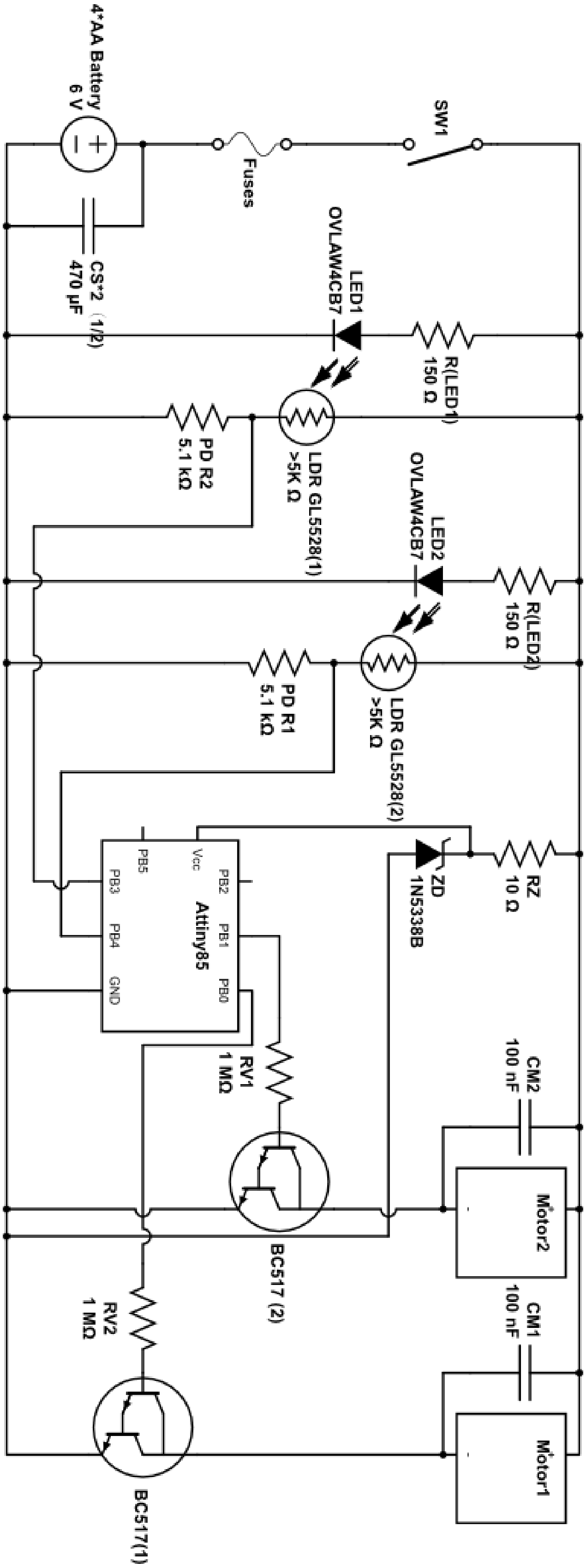
$$Vout = Vref \times \text{percentage of Duty Cycle}$$

<u>*Figure 7*</u>



<u>*Figure 8*</u>
Circuit Diagram

3 https://www.arduino.cc/en/Tutorial/PWM PWM



4*AA Battery
6 V
SW1
Fuses
LED1
OVLAW4CB7
CS*2 (1/2)
470 µF
R(LED1)
150 Ω
LDR GL5528(1)
>5K Ω
PD R2
5.1 kΩ
LED2
OVLAW4CB7
R(LED2)
150 Ω
PD R1
5.1 kΩ
LDR GL5528(2)
>5K Ω
RZ
10 Ω
ZD
1N5338B
Attiny85
PB2
PB1
PB0
Vcc
PB5
PB3
PB4
GND
CM2
100 nF
Motor2
RV1
1 MΩ
BC517 (2)
CM1
100 nF
RV2
1 MΩ
Motor1
BC517(1)

The circuit design is inspired from the high level idea written above and the circuit used last term (Appendix. Stage 1 chosen circuit). Several extra components have been added to achieve the behavior the bug requires to perform efficiently and safely.

Capacitor (CS (1/2)): Two Electrolytic Radial capacitor with 470uF (35V).
Fuse: The two fuses have been built into the circuit at 1st stage to protect the battery;
LED (1/2): The light source for the LDRs. With maximum power dissipation of *100mW*, maximum average forward current *25mAdc* (Appendix. Table for Max LED);
LDR (1/2): The resistance of LDR will increase as the intensity of light on it decreases.
The minimum resistance is *5kΩ*.
The maximum resistance is *200kΩ*, both of them are light resistances (Appendix. LDR data);
Atttiny85: Maximum operating voltage is *6Vdc*.
Maximum *40mAdc* I/O DC current per pin.
Maximum DC current Vcc to GND *200mdc* (Appendix. Maximum operation of Attiny85).
Zener Diode (ZD): Zener voltage should typically be *5.1Vdc*.
Maximum Zener resistance it can supply is *400Ω*.
The maximum current that can pass through is *0.930Adc*.
The maximum power dissipation is *5W* (Appendix. Zener diode Max).
BC517 (1/2): It is an NPN Darlington pair with minimum h_{FE}(current gain) =*30,000*. $V_{be(sat)}$(Saturation base-emitter voltage) = *1.4Vdc*.
$V_{ce(sat)}$(Saturation collector-emitter voltage) = *1Vdc*.
Maximum collector-emitter voltage is *30Vdc*.
Maximum base-emitter voltage is *10Vdc* (Appendix. BC517).
Motor (1/2): This is a 3-12V DC motor. With maximum load voltage 6V and maximum load current of 70-120mA.
Capacitor (CM (1/2)): Two ceramic capacitors with 100nF.

3.2 Circuit description with prediction and analysis

Overview of the circuit
(Please read this with Figure 8)

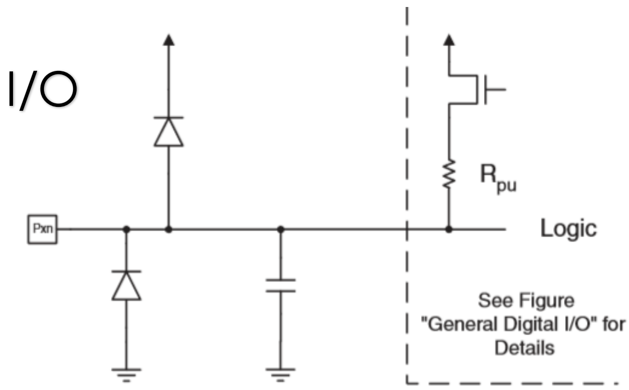
We use a digital circuit as the digital circuit itself will be smaller and simpler than an analog one as it is built on an integrated circuit. A simple circuit means it will have less internal effect (The whole circuit is built on a breadboard - all the parallel nodes will have a capacitance between each other and all the wires will have inductance - hence, the effect between them will be huge).

The whole circuit will then be divided into two parts: the input part and the output part. This idea is derived from the last term's circuit (Appendix. Stage 1 chosen circuit). On the previous circuit, the LDR shares a voltage with a potential divider. The voltage across the potential divider will be taken to

provide Vbe of the Darlington pair. The Darlington pair controls the current through the motor. The motor voltage is proportional to the motor speed (Appendix. Motor characteristics). As a result, when the resistance of LDR changes, the potential divider can be used to change the voltage drop across itself. This seems like a change to the sensitivity of the LDR. As we have a microcontroller, we can use the code to adjust the sensitivity of LDRs. At the input stage, the circuit directly can be a resistor-LDR potential divider.
The potential difference will be the input of the microcontroller. At all the pins, there is an internal protection circuit with high impedance (Figure 8). The input current can be assumed to be weak enough. The signal will pass through a circuit (Figure 5), which will convert the analog input into a digital value.

At output stage, we use PWM as the digital-to-analog convertor. As the maximum input or output current of any microcontroller pin is 40mA, the output pin cannot be directly connected to the motor used. The current the motor needs to achieve nearly full speed will be approximate 50mA. The NPN Darlington in this case can be use like a logic gate. Refer to the inside of the microcontroller as shown in Figure 8.

Figure 9



(User booklet of Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash, 10. I/O ports)

In this case, the RV (1/2) will share a voltage with the Rpu. The total voltage output will be 5V or 0V. The output voltage will be across the RV (1/2) and the rbe (1/2) of NPN Darlington pairs as a Darlington pair of CE can fully conduct when it is in saturation stage. The NPN Darlington pairs now can act as logic gates: 5V means on, and 0V is off. The output of the PWM can be transferred to the motors safely.

To make the system more stable, CS (1/2) and ZD are used. The two motors will make the current in the entire circuit unstable, which will make the voltage of the power supply unstable. A large capacitor across the power supply will prevent the abrupt change in the voltage of the power supply.

The Zener diode is used as a voltage rectifier, in series with a resistor, to provide a constant voltage of 5.1 Volts to the Vcc – the Vcc being used as the reference voltage of input and output.

Prediction and analysis:
For the R (LED (1/2)):
The voltage across the LEDs are set to be 4V.

$$\because i_{Forward\ average} = 25mA$$
$$\therefore R_{LED(1/2)} = \frac{V_{supply} - V_{LED}}{i_{Forward\ average}} = \frac{(6-4)v}{25mA} = 80\Omega$$

To be safety R=150Ω

For the PD R (1/2):
The current passing through it will never exceed 1.2mA.

$$i = \frac{V_{supply}}{R_{LDE\ min}} = \frac{6v}{5k\Omega} = 1.2\text{mA}$$

The PD R (1/2) here is to deduce the voltage level of the input. It can be set to be 5.1KΩ, which is closer to the minimum resistance of the LDR.
For RZ:

$$I_{max} = 930mA$$
$$\therefore R_z = \frac{V_{supply} - V_z}{i_{max}} = \frac{(6-5.1)v}{930mA} = 0.968\Omega$$
$$R_{\max zener\ resistance} = 400\Omega$$

Try 10Ω > 0.968Ω;

$$V_z = \frac{R_{\max zener\ resistance}}{R_{\max zener\ resistance} + 10\Omega} \times V_{supply} = \frac{400\Omega}{(400+10)\Omega} \times 6v = 5.85\text{v} > 5.1\text{v}$$

RZ=10Ω.

For RV (1/2):

$$V_{out} = 5V$$
$$V_{Rv} = V_{out} - V_{be} = 5 - 1.4 = 3.6v$$
$$\beta = 30000$$
$$I_c = I_{motor} = 0.05A$$
$$I_b = \frac{I_c}{\beta} = \frac{0.05}{30000} = 1.667 \times 10^{-6}\text{A}$$
$$R_v = \frac{V_{Rv}}{I_b} = \frac{3.6v}{1.667 \times 10^{-6}\text{A}} = 2.16M\Omega$$

1MΩ is used to provide enough base current.

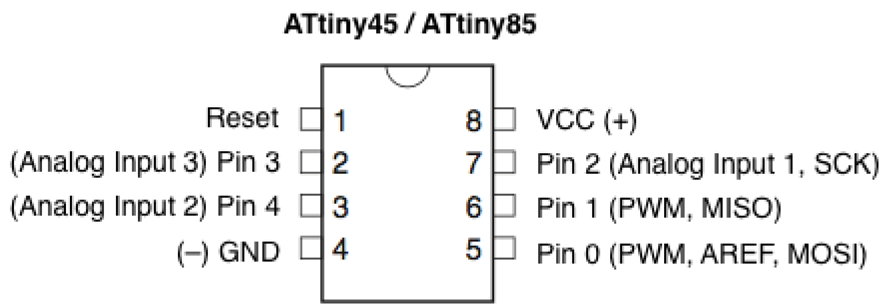
3.3 Flow chart and brief description of code

Function explain:

First of all, the input and output pins are figured out with "pinMode ()". The pins used as input have high-impedance state which is equivalent to a series resistor of 100MΩ in front of the pin. This indicates that very little current is required for the input pins and any small change in input can be amplified. Relatively, the output pins have low-impedance state which means that they can give enough amount of current (about 40 mA) to the circuit[4].

Pin 0 and pin 1 are set as output pins, while, pin 3 and pin 4 are set as input pins.

Figure 10



Source: http://highlowtech.org/?p=1695

At the beginning of the movement, both of the output pins are set to give a maximum voltage. With using "*analogWrite()*", the pin can generate a steady square wave of the stated duty cycle between 0 to 255. [5]

The "*analogRead()*" can read the voltage from the input pin and the minimum reading is 49mV per unit. It converts the voltage (0 – 5V) to an integer value (0 – 1023). If the difference of two inputs is not in the range that we defined, it will be amplified and the corresponding side of motor will be decelerated by the smaller output. The "do" loop will be repeated to control the bug following the line, until the difference of inputs is in the range, which means that the first part has been completed.

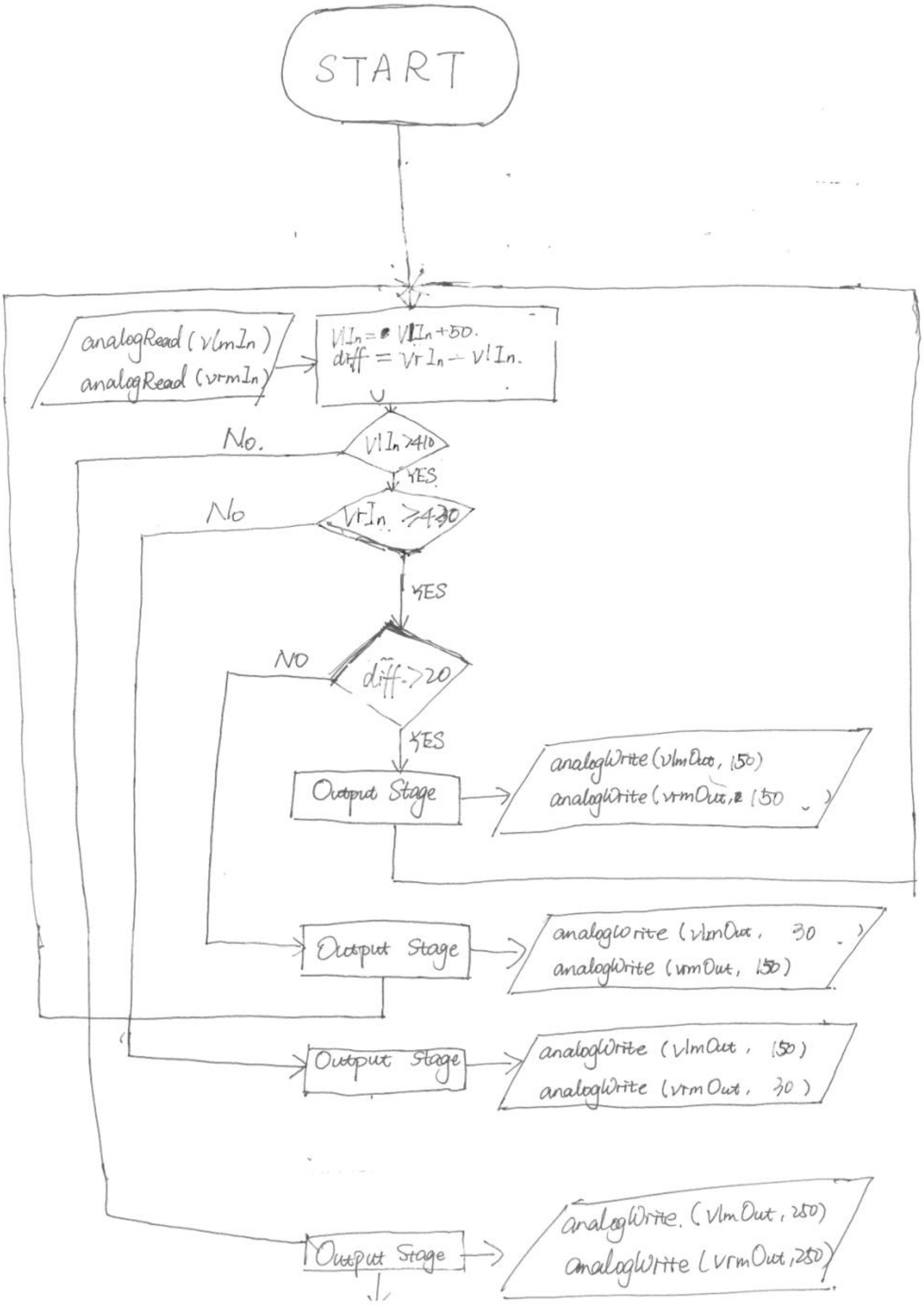
The "*delay()*" pauses the system from doing the next part for a fixed period of time and the number in the bracket is in milliseconds.

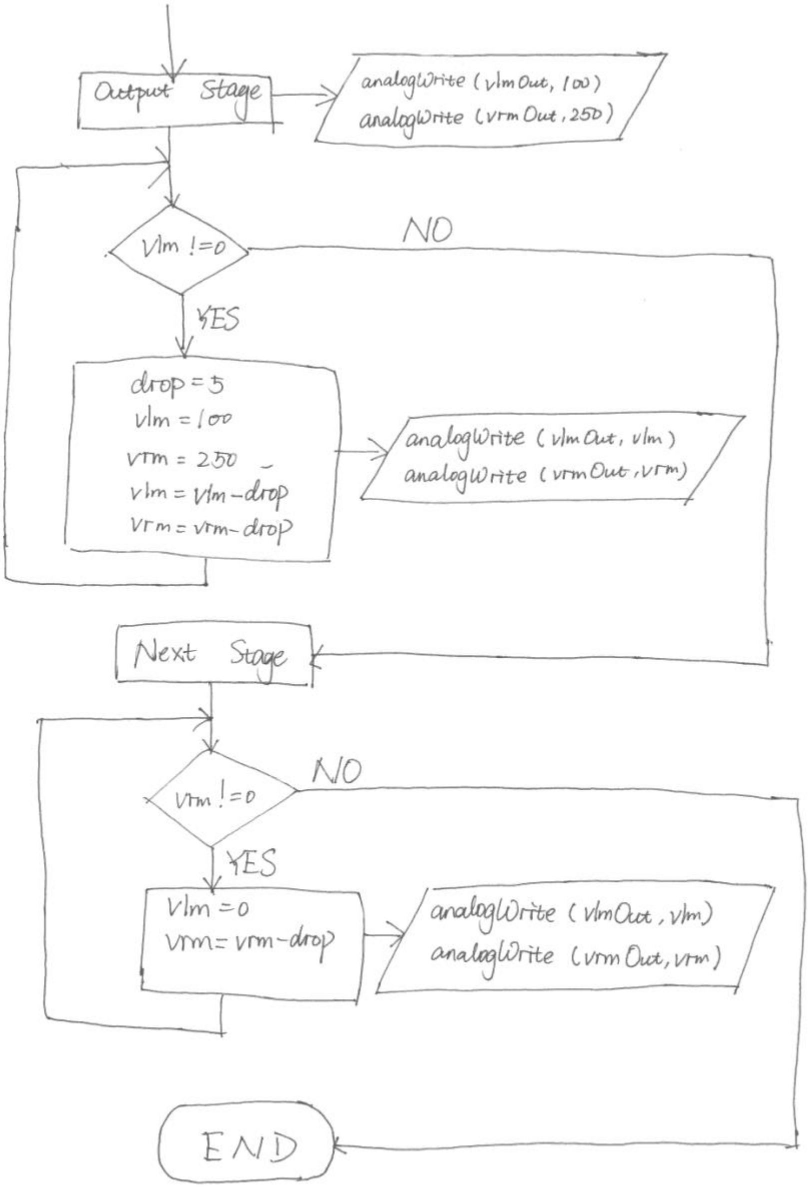
The whole program is ended with an infinite empty loop, which is the end part.

Figure 11
Logic diagram

[4] https://www.arduino.cc/en/Tutorial/DigitalPins)

[5] https://www.arduino.cc/en/Reference/AnalogWrite


START
analogRead (vlmIn)
analogRead (vrmIn)
VlIn = VlIn+50.
diff = VrIn − VlIn.
No.
VlIn>410
YES
No
VrIn >430
YES
NO
diff >20
YES
Output Stage
analogWrite(vlmOut, 150)
analogWrite(vrmOut, 150)
Output Stage
analogWrite (vlmOut, 30)
analogWrite (vrmOut, 150)
Output Stage
analogWrite (vlmOut, 150)
analogWrite (vrmOut, 30)
Output Stage
analogWrite. (vlmOut, 250)
analogWrite (vrmOut, 250)



Description of code (Appendix. Code final)

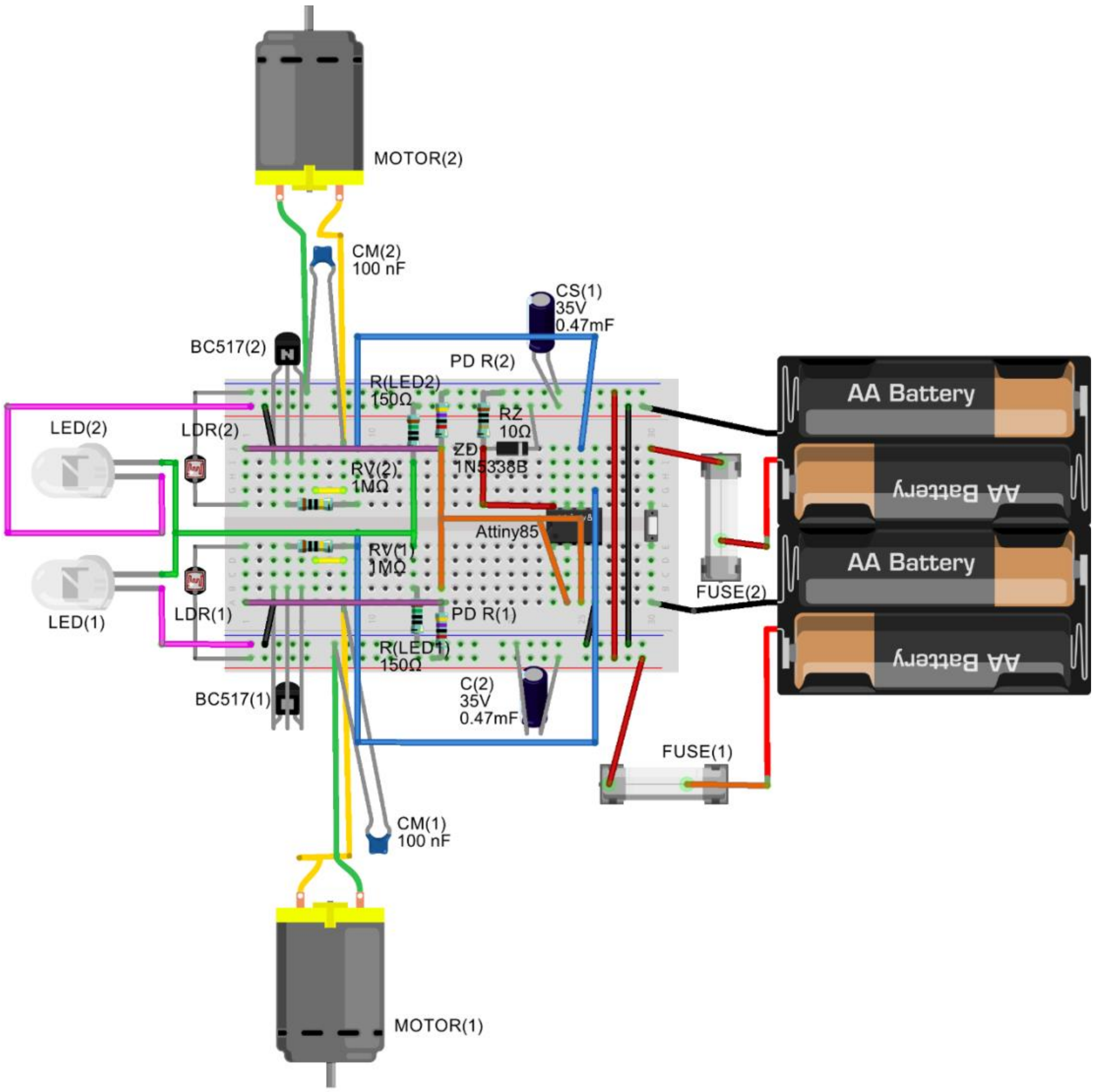
By checking the initial voltage across the potential divider, an offset value is given to the left sensor input. When the difference of two inputs is larger than 20 (analogue: 0.1V), the left sensor is over the black path and the right one is outside the path. This is considered as our reference position. When the input of right sensor is greater than 430 (analogue: 2.1V), the sensor is out of the black path and the difference is greater than 20, hence the bug is moving in a correct direction setting the outputs to give out the same value. With a difference smaller than 20, part of the left sensor should have moved away from the path, thus the left motor will be changed to a lower speed than the other one in order to turn the bug back. If the right input is smaller than 430, part of the right sensor should have been over the black path, giving a lower speed to the right output in order to turn the bug back to the reference position. The first part finishes and we switch to the second function when a value from the left sensor is greater than 410, which indicates that the line has completely faded.

For the second part, two outputs are given a nearly maximum voltage again. Then the time needed to run 10cm distance can be calculated by measuring the diameter and the speed of revolution of the bug wheel, which is about 640ms. So the bug will run with a high speed for 10cm.

For the final part, different initial voltages are set, but they both decrease by the same rate until the one with the relatively smaller initial value becomes zero. Then the other motor continues to reduce its respective voltage until it becomes zero and the bug finally stops.

Figure 12
Layout diagram

The PD R (1/2) in the circuit should be 5.1Ω



4. EEBUG enhancements and testing

(Extra for functionality beyond design brief)

4.1 Test instructions for setting-up the bug

As the microcontroller has 2 extra pins, two extra function can be added to the bug.

A special switch

It is possible to add a *do* loop into the code which will allow the whole code to run. The condition of this can be used to set an on or off switch. From the energy conversion module studied this term, the current carrying coils will have mutual inductance with each other. The load in secondary coil can be copied to primary coil meaning the load is replaced by a special circuit which records a number (binary with 010101). This information can be transferred back to the primary coil. As a result, we can add the primary coil into the circuit and build the secondary coil into a card. When the card is close enough to the primary coil, it will be activated and transfer the number to the primary coil. Before the *do* loop, an *if* loop will be set up. If we receive the number, set a = 1. Otherwise check the *if* statement repeatedly. The condition of the *do* loop will be while a = 1. With this switch, our bug can only be accessed by the card made. This is similar to the method used in oyster cards.

Music at the end:

For the other pin, it can be connected to a speaker which has a sound recording in it. At the end of the code, it will go into an infinite loop which will active the speaker to make the sound. This will signify the end of the whole program.

4.2 Test instructions for setting-up the bug

The coding part

While programming the microcontroller, several problems were faced. Details of the original (first) code written and tested, followed by an improved code and lastly, the final code used are provided below.

<u>For the first code version</u> (Refer to Appendix on Code1)

At the beginning of the movement, both of the output pins are set to give a maximum voltage. Taking inputs from two sensors, if the difference of two inputs is not in the bandwidth that we defined, it will be amplified and the corresponding side of the motor will be decelerated by the smaller output. The *do* loop will repeat to control and check the bug following the line until the difference of inputs is in the bandwidth, which means that the first part has been completed.

For the second part, the motors are set to the maximum voltage again. The diameter and the speed of revolution of the bug wheel are measured, helping us to calculate the time needed to run 10cm distance that is about 640ms.

For the final part, two different voltage values are initialized and are decreased at the same rate until the one with smaller initial value reaches zero. The larger value decreases to zero and the big eventually comes to a halt.

Shortcoming:
The bug does follow the first part of the journey. It only does part two and three directly because of the complex conditions of the loop.
The input pin setting at the beginning is wrong.

<u>For the second code version</u> (Refer to Appendix onCode2)
The wrong input setting has been fixed. The inputs have an obvious difference even under the same condition, which is the reason for the complex condition. Therefore, to start with, each time when the inputs are taken, an additional value of 50 (analogue: 0.244V) will be added to the left input in order to give the same value as the right one under the same condition.

"diff" is defined as the right sensor input voltage minus the left sensor input voltage. If "diff" is positive, the right sensor moves out of the line, so a larger output will be given to the right motor than the other one. Similarly, for a negative "diff" value the left motor will be set faster than the right motor by the output. When the "diff" is greater than 15 (0.073V), the first part has finished and the program will go to the next part to control the bug running in straight line.

Shortcoming:
The bug can follow the line but it is unable to reach the condition that terminates the first part.

<u>For the finial code version (Refer to Appendix on Code final)</u>
At the first stage, the program keeps taking inputs and implements corresponding instructions to make sure that the left sensor is always over the black line while the right sensor stays outside the line. When the left sensor completely detects the white board, the second stage begins. Equal initial high voltage is given to each output to let the bug run in a straight line. After the certain time passes for the bug to cover the required 10cm distance, the third and final stage begins. The two motors are set to have the same decreasing rate but different initial values. Thus the bug draws a spiral with decreasing radius, until both of the outputs become zero and the bug comes to a halt.

The circuit part
<u>The higher reference voltage</u>:
After setting up the circuit, a testing code is used to test if the circuit works properly. The testing code only checks if the bug moves forward in a straight

line with constant speed. A 1Ω resistor is in series with each motor for checking the motor current. The current through the left motor is 0.0573A and the current passing through the right motor is 0.0656A. The Vbe of the Darlington pairs is 1.2689V for left and 1.2675V for right.

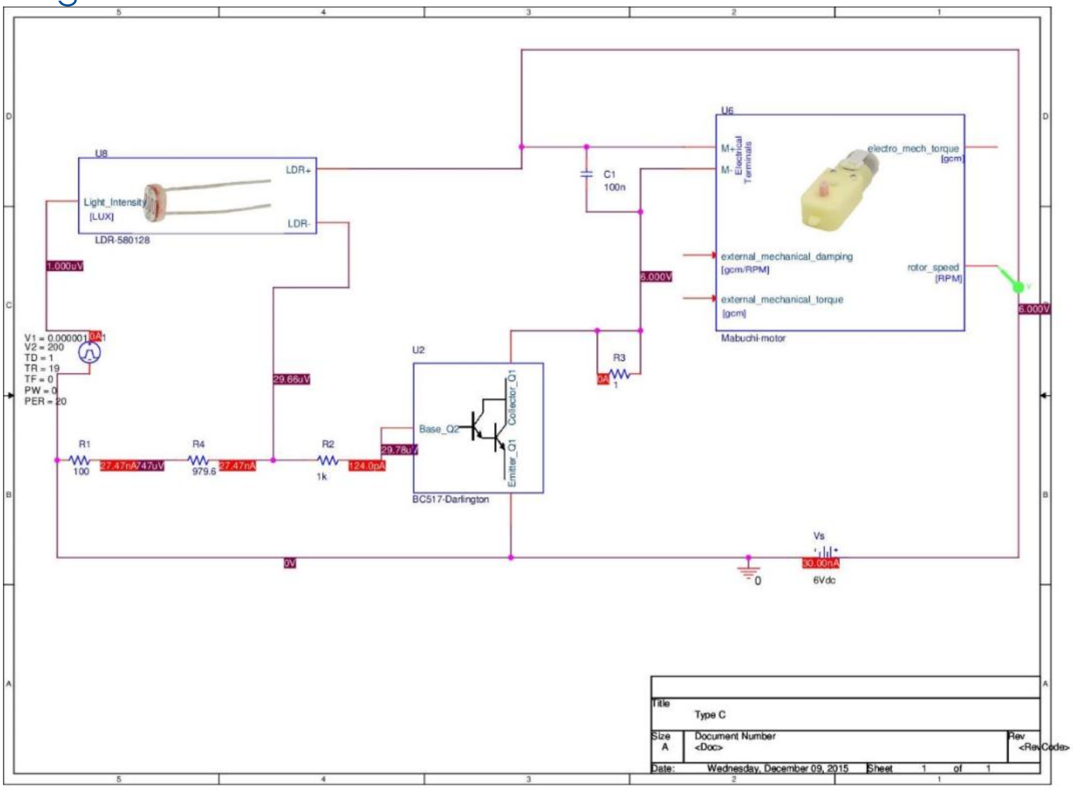
All the data is fine and the bug works properly.

Even though the code is written to only allow the bug to go 10cm forward is, it always goes an extra 2-3cm forward. The Vout (checked on the DVM) shows it exceeds 5V. A Zener diode is added to the circuit to regulate the Vcc to 5.1V. After implementing this change, the distance it covers is slightly less than 10cm, but it is within a small range.

<u>The effect of the motor on the circuit:</u>
As from last term's circuit, there is a 100nF capacitor across the motor. Initially, no extra capacitors were added to the circuit. On using the testing code, the bug worked fine. Then, an extra step was added into the code to make it move in a spiral. The bug was to stop at the end, however, it didn't. It makes a spiral for a few seconds and then starts to move in a straight line again. All the chips have a reset voltage level. The phenomena of our bug shows that the chip was reset, making the bug restart the entire program again. This indicates the instability of the power supply, making the voltage rectifier not work properly. During testing, the voltage across the power supply (+ive and –ive terminals) showed a triangle-like wave instead of a horizontal, constant line (indicating constant voltage). From the energy conversion module, it was studied that a DC motor will generate a back-E.M.F. when it is running. The E.M.F. generated will be proportional to the motor speed. When the speed changes, the E.M.F. changes. This will cause a change in the current. As the power supply is supplying a constant power(P=IV), the voltage will change. In conclusion, larger capacitors are added to the circuit across the power supply to prevent this abrupt change in the voltage of power supply.

Appendices:

Stage 1 chosen circuit



R4 is the potential divider, R1 is for current limit.

Table for Max LED

Absolute Maximum Ratings (T_A = 25° C unless otherwise noted)

Storage Temperature Range	-40 ~ +100°C
Operating Temperature Range	-40 ~ +100°C
Reverse Voltage	5 V
Power Dissipation	100 mW
Average Forward Current	25 mA
Peak Forward Current (Duty Ratio = 1/10, Pulse Width =0.1ms)	100 mA
Current Linearity vs Ambient Temperature	-0.29 mA/°C
LED Junction Temperature	125°C
Electrostatic Discharge Classification (JEDEC-JESD22-A114F)	Class 1C
Lead Soldering Temperature (5 seconds maximum)	260°C

LDR data

GL55 Series CdS Photoresistor Manual

Specification	Light resistance (10Lux) (KΩ)	Dark resistance (MΩ)	γ_{10}^{100}	Response time (ms)		Illuminance resistance Fig. No.
				Increase	Decrease	
Φ5 series	5-10	0.5	0.5	30	30	2
	10-20	1	0.6	20	30	3
	20-30	2	0.6	20	30	4
	30-50	3	0.7	20	30	4
	50-100	5	0.8	20	30	5
	100-200	10	0.9	20	30	6

Test Conditions

Max. external voltage: Maximum voltage to be continuously given to component in the dark.

Dark resistance: Refer to the resistance ten seconds after the 10Lux light is shut up.

Max. power consumption: Maximum power at the environmental temperature 25℃.

Light resistance: Irradiated by 400-600Lux light for two hours, then test with 10Lux under standard light source A(as colour temperature 2856K).

γ value: Logarithm of the ratio of the standard resistance value under 10Lux and that under 100Lux.

$$\gamma = \frac{Lg(R10/R100)}{Lg(100/10)} = Lg(R10/R100)$$

R10,R100 are the resistances under 10Lux and 100Lux respectively.

Maximum operation of Attiny85

21. Electrical Characteristics

21.1 Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{RESET}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{RESET}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Zener diode Max

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Max. Steady State Power Dissipation @ T_L = 25°C, Lead Length = 3/8 in Derate above 25°C	P_D	5 40	W mW/°C
Junction−to−Lead Thermal Resistance	θ_{JL}	25	°C/W
Operating and Storage Temperature Range	T_J, T_{stg}	−65 to +200 (Note 1)	°C

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

1. Max operating temperature for DC conditions is 150°C, but not to exceed 200°C for pulsed conditions with low duty cycle or non−repetitive.

Device† (Note 2)	Device Marking	Zener Voltage (Note 3) V_Z (Volts) Min	Nom	Max	@ I_{ZT} mA	Zener Impedance (Note 3) Z_{ZT} @ I_{ZT} Ω	Z_{ZK} @ I_{ZK} Ω	I_{ZK} mA	Leakage Current I_R @ V_R μA Max	Volts	I_R (Note 4) A	ΔV_Z (Note 5) Volts	I_{ZM} (Note 6) mA
1N5333B	***1N5333B***	***3.14***	***3.3***	***3.47***	***380***	***3***	***400***	***1***	***300***	***1***	***20***	***0.85***	***1440***
1N5334B	1N5334B	3.42	3.6	3.78	350	2.5	500	1	150	1	18.7	0.8	1320
1N5335B	1N5335B	3.71	3.9	4.10	320	2	500	1	50	1	17.6	0.54	1220
1N5336B	1N5336B	4.09	4.3	4.52	290	2	500	1	10	1	16.4	0.49	1100
1N5337B	***1N5337B***	***4.47***	***4.7***	***4.94***	***260***	***2***	***450***	***1***	***5***	***1***	***15.3***	***0.44***	***1010***
1N5338B	***1N5338B***	***4.85***	***5.1***	***5.36***	***240***	***1.5***	***400***	***1***	***1***	***1***	***14.4***	***0.39***	***930***
1N5339B	***1N5339B***	***5.32***	***5.6***	***5.88***	***220***	***1***	***400***	***1***	***1***	***2***	***13.4***	***0.25***	***865***
1N5340B	1N5340B	5.70	6.0	6.30	200	1	300	1	1	3	12.7	0.19	790
1N5341B	***1N5341B***	***5.89***	***6.2***	***6.51***	***200***	***1***	***200***	***1***	***1***	***3***	***12.4***	***0.1***	***765***
1N5342B	***1N5342B***	***6.46***	***6.8***	***7.14***	***175***	***1***	***200***	***1***	***10***	***5.2***	***11.5***	***0.15***	***700***
1N5343B	1N5343B	7.13	7.5	7.88	175	1.5	200	1	10	5.7	10.7	0.15	630
1N5344B	1N5344B	7.79	8.2	8.61	150	1.5	200	1	10	6.2	10	0.2	580
1N5345B	1N5345B	8.27	8.7	9.14	150	2	200	1	10	6.6	9.5	0.2	545
1N5346B	1N5346B	8.65	9.1	9.56	150	2	150	1	7.5	6.9	9.2	0.22	520
1N5347B	***1N5347B***	***9.50***	***10***	***10.5***	***125***	***2***	***125***	***1***	***5***	***7.6***	***8.6***	***0.22***	***475***

Devices listed in ***bold, italic*** are ON Semiconductor **Preferred** devices. **Preferred** devices are recommended choices for future use and best overall value.

2. **TOLERANCE AND TYPE NUMBER DESIGNATION:** The JEDEC type numbers shown indicate a tolerance of ±5%.
3. **ZENER VOLTAGE (V_Z) and IMPEDANCE (I_{ZT} and I_{ZK}):** Test conditions for zener voltage and impedance are as follows: I_Z is applied 40 ±10 ms prior to reading. Mounting contacts are located 3/8″ to 1/2″ from the inside edge of mounting clips to the body of the diode (T_A = 25°C +8°C, −2°C).
4. **SURGE CURRENT (I_R):** Surge current is specified as the maximum allowable peak, non−recurrent square−wave current with a pulse width, PW, of 8.3 ms. The data given in Figure 5 may be used to find the maximum surge current for a square wave of any pulse width between 1 ms and 1000 ms by plotting the applicable points on logarithmic paper. Examples of this, using the 3.3 V and 200 V zener are shown in Figure 6. Mounting contact located as specified in Note 2 (T_A = 25°C +8°C, −2°C).
5. **VOLTAGE REGULATION (ΔV_Z):** The conditions for voltage regulation are as follows: V_Z measurements are made at 10% and then at 50% of the I_Z max value listed in the electrical characteristics table. The test current time duration for each V_Z measurement is 40 ±10 ms. Mounting contact located as specified in Note 2 (T_A = 25°C +8°C, −2°C).
6. **MAXIMUM REGULATOR CURRENT (I_{ZM}):** The maximum current shown is based on the maximum voltage of a 5% type unit, therefore, it applies only to the B−suffix device. The actual I_{ZM} for any device may not exceed the value of 5 watts divided by the actual V_Z of the device. T_L = 25°C at 3/8″ maximum from the device body.

†The "G'' suffix indicates Pb−Free package or Pb−Free packages are available.

Device† (Note 2)	Device Marking	Zener Voltage (Note 3) V_Z (Volts) Min	Nom	Max	@ I_{ZT} mA	Zener Impedance (Note 3) Z_{ZT} @ I_{ZT} Ω	Z_{ZK} @ I_{ZK} Ω	I_{ZK} mA	Leakage Current I_R @ V_R µA Max	Volts	I_R (Note 4) A	ΔV_Z (Note 5) Volts	I_{ZM} (Note 6) mA
1N5333B	***1N5333B***	***3.14***	***3.3***	***3.47***	***380***	***3***	***400***	***1***	***300***	***1***	***20***	***0.85***	***1440***
1N5334B	1N5334B	3.42	3.6	3.78	350	2.5	500	1	150	1	18.7	0.8	1320
1N5335B	1N5335B	3.71	3.9	4.10	320	2	500	1	50	1	17.6	0.54	1220
1N5336B	1N5336B	4.09	4.3	4.52	290	2	500	1	10	1	16.4	0.49	1100
1N5337B	***1N5337B***	***4.47***	***4.7***	***4.94***	***260***	***2***	***450***	***1***	***5***	***1***	***15.3***	***0.44***	***1010***
1N5338B	***1N5338B***	***4.85***	***5.1***	***5.36***	***240***	***1.5***	***400***	***1***	***1***	***1***	***14.4***	***0.39***	***930***
1N5339B	***1N5339B***	***5.32***	***5.6***	***5.88***	***220***	***1***	***400***	***1***	***1***	***2***	***13.4***	***0.25***	***865***
1N5340B	1N5340B	5.70	6.0	6.30	200	1	300	1	1	3	12.7	0.19	790
1N5341B	***1N5341B***	***5.89***	***6.2***	***6.51***	***200***	***1***	***200***	***1***	***1***	***3***	***12.4***	***0.1***	***765***
1N5342B	***1N5342B***	***6.46***	***6.8***	***7.14***	***175***	***1***	***200***	***1***	***10***	***5.2***	***11.5***	***0.15***	***700***
1N5343B	1N5343B	7.13	7.5	7.88	175	1.5	200	1	10	5.7	10.7	0.15	630
1N5344B	1N5344B	7.79	8.2	8.61	150	1.5	200	1	10	6.2	10	0.2	580
1N5345B	1N5345B	8.27	8.7	9.14	150	2	200	1	10	6.6	9.5	0.2	545
1N5346B	1N5346B	8.65	9.1	9.56	150	2	150	1	7.5	6.9	9.2	0.22	520
1N5347B	***1N5347B***	***9.50***	***10***	***10.5***	***125***	***2***	***125***	***1***	***5***	***7.6***	***8.6***	***0.22***	***475***

Devices listed in ***bold, italic*** are ON Semiconductor **Preferred** devices. **Preferred** devices are recommended choices for future use and best overall value.

2. **TOLERANCE AND TYPE NUMBER DESIGNATION:** The JEDEC type numbers shown indicate a tolerance of ±5%.
3. **ZENER VOLTAGE (V_Z) and IMPEDANCE (I_{ZT} and I_{ZK}):** Test conditions for zener voltage and impedance are as follows: I_Z is applied 40 ±10 ms prior to reading. Mounting contacts are located 3/8″ to 1/2″ from the inside edge of mounting clips to the body of the diode (T_A = 25°C +8°C, −2°C).
4. **SURGE CURRENT (I_R):** Surge current is specified as the maximum allowable peak, non−recurrent square−wave current with a pulse width, PW, of 8.3 ms. The data given in Figure 5 may be used to find the maximum surge current for a square wave of any pulse width between 1 ms and 1000 ms by plotting the applicable points on logarithmic paper. Examples of this, using the 3.3 V and 200 V zener are shown in Figure 6. Mounting contact located as specified in Note 2 (T_A = 25°C +8°C, −2°C).
5. **VOLTAGE REGULATION (ΔV_Z):** The conditions for voltage regulation are as follows: V_Z measurements are made at 10% and then at 50% of the I_Z max value listed in the electrical characteristics table. The test current time duration for each V_Z measurement is 40 ±10 ms. Mounting contact located as specified in Note 2 (T_A = 25°C +8°C, −2°C).
6. **MAXIMUM REGULATOR CURRENT (I_{ZM}):** The maximum current shown is based on the maximum voltage of a 5% type unit, therefore, it applies only to the B−suffix device. The actual I_{ZM} for any device may not exceed the value of 5 watts divided by the actual V_Z of the device. T_L = 25°C at 3/8″ maximum from the device body.

†The “G” suffix indicates Pb−Free package or Pb−Free packages are available.

Motor characteristics

EEbug Type C					
L	(28^2*13)/L^2/lux	Motor speed/rpm	Vmotor/V	Motor speed/rpm	Motor current/A
120	0.707777778	0	0.098	0	0.002
118	0.731973571	0	0.569	0	0.007
116	0.757431629	0	1.136	0	0.01
114	0.784241305	0	1.671	0	0.015
112	0.8125	1512	2.521	1512	0.022
110	0.84231405	3312	3.212	3312	0.029
108	0.873799726	3888	3.8	3888	0.035
106	0.907084372	5328	3.883	5328	0.045
104	0.942307692	5976	4.515	5976	0.047
102	0.979623222	7272	4.7	7272	0.048
100	1.0192	7920	4.86	7920	0.051
98	1.06122449	8280	4.875	8280	0.053
96	1.105902778	9000	4.882	9000	0.055
94	1.153463105	9000	4.888	9000	0.056
92	1.20415879	9000	4.891	9000	0.056
90	1.258271605	9000	4.891	9000	0.057
88	1.316115702	9000	4.896	9000	0.057
86	1.378042185	9000	4.902	9000	0.057
84	1.444444444	9000	4.908	9000	0.057
82	1.515764426	9000	4.913	9000	0.058
80	1.5925	9000	4.913	9000	0.058

Type E

L/cm	s rev/s	Motor speed Sm in rpm	V+/V	Vb/V	Vmotor/V	Vmotor curent/V	imotor/A
7	0.75	9577	5.826	-0.722	5.672	0.05689	0.05689
9	0.75	9639	5.797	0.726	5.651	0.05706	0.05706
19	0.75	9628	5.735	-0.731	5.587	0.05932	0.05932
29	0.77	9385	5.701	-0.721	5.543	0.05858	0.05858
40	0.77	9410	5.685	-0.717	5.53	0.05518	0.05518
50	0.77	9387	5.681	-0.712	5.51	0.05537	0.05537
60	0.77	9365	5.681	-0.709	5.502	0.05453	0.05453
70	0.77	9300	5.679	-0.703	5.473	0.0552	0.0552
80	0.78	9292	5.675	-0.702	5.456	0.0548	0.0548
90	0.79	9174	5.673	-0.698	5.407	0.05472	0.05472
100	0.78	9213	5.673	-0.698	5.406	0.05384	0.05384
110	0.89	8072	5.679	-0.674	4.774	0.0507	0.0507
120	0.14	6323	5.687	-0.654	4.339	0.04556	0.04556
130	0	0	5.696	-0.612	0.446	0.06733	0.06733

Code1

```
int IIn=4;
```

```
int rIn=3;
int vlmOut=0;
int vrmOut=1;
void setup() {
// put your setup code here, to run once:
pinMode (lIn,INPUT);
pinMode (rIn,INPUT);
pinMode (vlmOut,OUTPUT);
pinMode (vrmOut,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
// follow the line part
analogWrite (vlmOut, 250);
analogWrite (vrmOut, 250);
int vlIn, vrIn;
int diff,k,change;
int bandwith=41;
k=1;
vlIn=analogRead (lIn);
vrIn=analogRead (rIn);
do {
vlIn=analogRead (lIn);
vrIn=analogRead (rIn);
delay (1000);
diff=vlIn-vrIn;
if ( diff>bandwith ){
change=k*diff;
change=change*256/1024;
analogWrite (vlmOut, 250);
analogWrite (vrmOut, 0);
delay (100);
}
if ( diff<bandwith ){
change=-k*diff;
change=change*256/1024;
analogWrite (vlmOut, 0);
analogWrite (vrmOut, 250);
delay (100);
}
}while ( (((vlIn-vrIn<0) && (vlIn-vrIn>-bandwith)) || ((vlIn-v
rIn>0) && (vlIn-
vrIn<bandwith))) && ((vlIn>430) && (vrIn>430)) );
// straight line part
analogWrite (vlmOut, 250);
analogWrite (vrmOut, 250);
delay (640);     // run for 640ms
// the last part
```

```
analogWrite (vlmOut, 100);
analogWrite (vrmOut, 250);
int vlm=100;
int vrm=250;
int drop=5;
while (! vlm==0){
vlm=vlm-drop;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
delay (500);
}
while (! vrm==0){
vlm=0;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
delay(200);
}
while (1){}
}
```

Code2

```
int lIn=A2;
int rIn=A3;
int vlmOut=0;
int vrmOut=1;
```

```
void setup() {
// put your setup code here, to run once:
pinMode (vlmOut,OUTPUT);
pinMode (vrmOut,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
// follow the line part
int vlIn, vrIn;
int diff;
do {
vlIn=analogRead (lIn)+50;
vrIn=analogRead (rIn);
diff=vrIn-vlIn;
if (diff>0){
analogWrite (vlmOut, 30);
analogWrite (vrmOut, 150);
}
if (diff<0){
analogWrite (vlmOut, 150);
analogWrite (vrmOut, 30);
}
}while ( diff<15 );
// straight line part
analogWrite (vlmOut, 250);
analogWrite (vrmOut, 250);
delay (640);     // run for 640ms
// the last part
analogWrite (vlmOut, 100);
analogWrite (vrmOut, 250);
int vlm=100;
int vrm=250;
int drop=5;
while (! vlm==0){
vlm=vlm-drop;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
delay (500);
}
while (! vrm==0){
vlm=0;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
delay(200);
}
while (1){}
```

```
}
```

Code final

```
int lIn=A2;
int rIn=A3;
int vlmOut=0;
int vrmOut=1;
void setup() {
// put your setup code here, to run once:
pinMode (vlmOut,OUTPUT);
```

```
pinMode (vrmOut,OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
// follow the line part
int vlIn, vrIn;
int diff;
do {
vlIn=analogRead (lIn)+50;
vrIn=analogRead (rIn);
diff=vrIn-vlIn;
if (vrIn>=430){
if (diff>20){
analogWrite (vlmOut, 150);
analogWrite (vrmOut, 150);
}
else {
analogWrite (vlmOut, 30);
analogWrite (vrmOut, 150);
}
}
else {
analogWrite (vlmOut, 150);
analogWrite (vrmOut, 30);
}
}while ( vlIn>410 );
// straight line part
analogWrite (vlmOut, 250);
analogWrite (vrmOut, 250);
delay (640);     // run for 640ms
// the last part
analogWrite (vlmOut, 100);
analogWrite (vrmOut, 250);
int vlm=100;
int vrm=250;
int drop=5;
while (! vlm==0){
vlm=vlm-drop;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
delay (500);
}
while (! vrm==0){
vlm=0;
vrm=vrm-drop;
analogWrite (vlmOut,vlm);
analogWrite (vrmOut,vrm);
```

```
delay(200);
}
while (1){}
}
```