University of Nottingham
UK | CHINA | MALAYSIA

EEEE1002
Applied Engineering Design

Session 3 Introduction

EEEE1002: An Introduction

Very Important: Module Engagement

- The purpose of the module is to acquire practical engineering skills
- Engagement is a compulsory requirement to pass the module
- When scheduled to be in a lab
    - Be in the lab by 09:00 and finish at 17:00
    - Lunch break should be limited to 1 hour maximum, between 12:00 and 14:00
    - Attendance will be monitored throughout the day
- **Non engagement may be penalised, unless supported by an EC claim & tutor informed**
- **This module needs to be passed. It can only be resat in attendance the following year i.e. it will add a year to the duration of your degree**

EEEE1002: An Introduction

What is EEEE1002?

- It is a practical ‘hands on’ engineering design module.
- It is worth 40 credits, rule of thumb…
  - 1 Credit = 10 hours of effort $\therefore$ 40 credits = 400 hours
  - You will spend approximately 200 hours in the lab
  - Therefore significant effort required outside the lab
  - 200 hours in the lab, 200 hours outside the lab
- Non-engagement will attract a penalty
- It must be passed to progress to Year 2
- **The module can only be re-taken in attendance i.e. failing the module will delay your progression by one year**

EEEE1002: An Introduction

Module Aims

- Further develop your knowledge by putting into practise concepts covered in the lectures
- Develop Skills Required by Employers
  - Develops practical skills
  - Develops engineering ways of thinking
  - Develop effective approaches to project management
- Develop the ability to work independently
- Develop the ability to work effectively as a team

EEEE1002: An Introduction

Support throughout the module

- **3 x Academic staff running the module, offering in-lab and remote support:**
  - Steve Greedy, Sergiy Korposh and Adam Walker
- **4 x Teaching Assistants offering in-lab and remote support:**
  - Ahmed Aldabbagh, Richard Davies, Nat Dacombe, David Dewar
- **Technical staff as required**
  - Mark Birkin and Alex Ottway





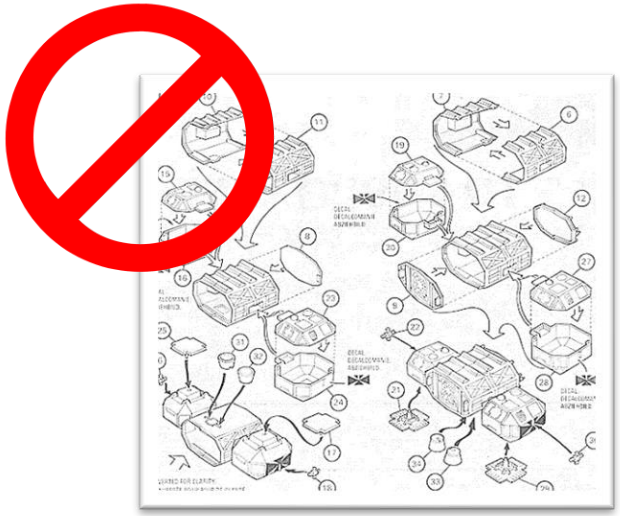


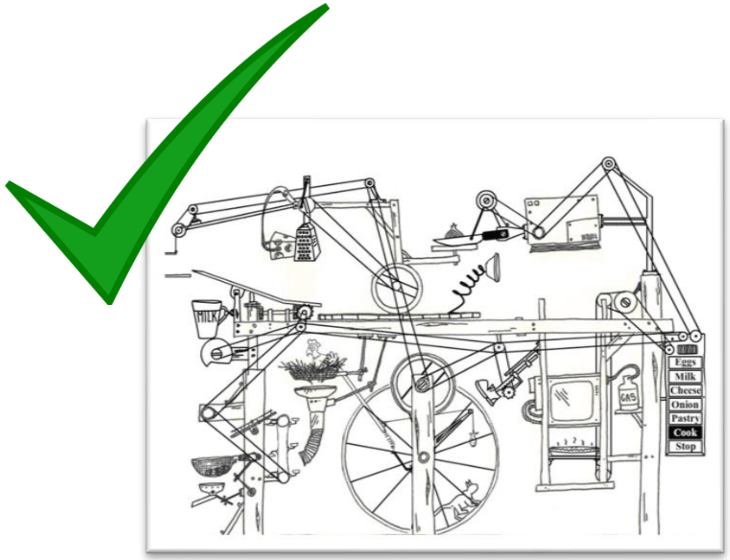


EEEE1002: An Introduction

Lots of support available, however

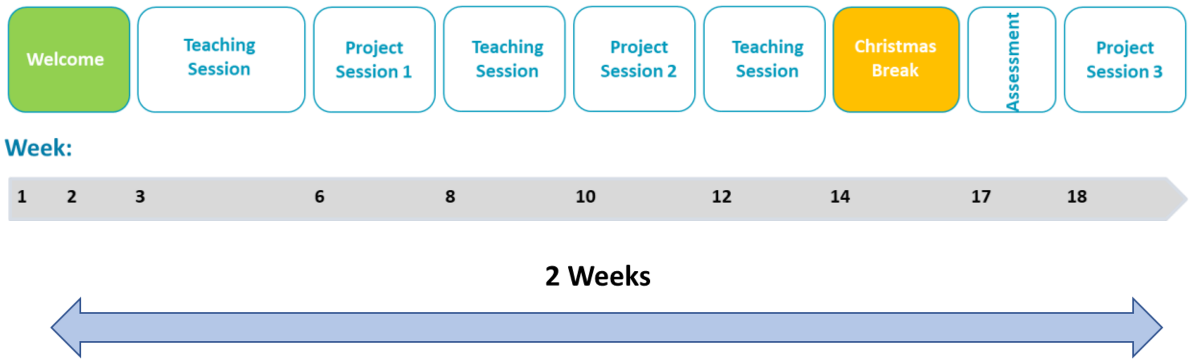
- This is not an ‘Instructional’ module
- No advance information (maybe a few hints…)
- We will help guide you in an Engineering approach
- We will not provide solutions





EEEE1002: An Introduction

Project Session Schedule



Project Session: First Week				
Practical Labs: Groups A and B – 09:00 to 17:00				
M	T	W	T	F
am	am	am	am	am
pm	pm		pm	pm
Self directed study: Groups C and D (plan for 32hrs)				
• Background research for EEEE1002 • 1-2-1 Personal tutorials • Revisit **all** modules: learning material and coursework requirements				

Project Session: First Week				
Practical Labs: Groups C and D – 09:00 to 17:00				
M	T	W	T	F
am	am	am	am	am
pm	pm		pm	pm
Self directed study: Groups A and B (plan for 32hrs)				
• Background research for EEEE1002 • 1-2-1 Personal tutorials • Revisit **all** modules: learning material and coursework requirements				

- Project sessions span **two weeks**. One of these weeks will be spent on practical work and one will be spent attending face-to-face technical tutorials
- **Groups A and B** will have practical work in the first week of the two week sessions.
- **Groups C and D** will have practical work in the second week of the two week sessions
- During a practical week you will spend **4.5 days** in the lab
- In the non-lab based week you should devote this to self directed study for an equivalent amount of time (~32 hours).

EEEE1002: An Introduction

Changes for this year

- The platform is again a 3 wheel, 2WD vehicle
- Individual and group work
- Introduction of an IoT approach to enable vehicle to anything communications V2X
- We will have a group photo

The end game remains the same

EEEE1002: An Introduction

Ultimate Goal: Development of an autonomous vehicle utilising computer vision to navigate

- SAE (Society of Automotive Engineers) Levels of automation:

SAE Level	Name	Narrative definition		Execution of steering and acceleration/ deceleration	Monitoring of driving environment
Human driver monitors the driving environment					
0	No Automation	The full-time performance by the human driver of all aspects of the dynamic driving task, even when "enhanced by warning or intervention systems"		Human driver	Human driver
1	Driver Assistance	The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration"	using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task	Human driver and system	
2	Partial Automation	The driving mode-specific execution by one or more driver assistance systems of *both steering and acceleration/deceleration*		System	
Automated driving system monitors the driving environment					
3	Conditional Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task	with the expectation that the *human driver will respond appropriately to a request to intervene*	System	System
4	High Automation		*even if a human driver does not respond appropriately to a request to intervene*		
5	Full Automation		*under all roadway and environmental conditions that can be managed by a human driver*		

https://en.wikipedia.org/wiki/Self-driving_car

- We will aim for SAE level 4 i.e. your vehicle should complete a course without human intervention

EEEE1002: An Introduction

On the route to autonomy: Sessions & Subsystems

1. Intro to test & measurement principles for electrical and electronic engineering and basic motion
2. Sensors & serial communications
3. **Optical line following**
4. Vehicle to anything (V2X) communications
5. Intro to computer vision
6. Big track challenge – incorporate one or more sub-systems to navigate a given route and complete technical challenges

- **Specific detail will be provided on the 1st Monday of each project session. Generic technical detail will be covered in the Monday seminars**

Session 3 Optical Line Following

Assessment:

- **Following session 4:**
  - **20%:** Individual report on detailing work undertaken in sessions 3 and 4
  - **5% :** Group poster detailing technical design of vehicle
- Final coursework descriptions/requirements released on the 25th February 2022
  - Keep a detailed log book!
- **Course work submission dates:**
  - **Poster: 3rd March 2022 at 15:00**
  - **Report: 10th March 2022 at 15:00**

Session 3 Optical Line Following

Project Log (session 2 onwards)

- Throughout this module you will be required to produce technical reports on the work you have undertaken
- It is therefore even more important from now on that you keep a log of the work you undertake, information to log can include:
  - Written notes that capture your approach/thinking
  - Circuit diagrams
  - Measurement data
  - Screen shots (hint: use PRINTSCRN or ALT+PRINTSCRN to capture screen shots or the active window and paste into a document, or paste into Paint and save as an image)
  - Photos
  - Key references (web links, .PDFs etc)

Session 3 Optical Line Following

Introduction to Session 3

- Session 3 will
  - Use an optical sensor system to implement a line following system for the EEE-Bot
  - Optical line following is a very common activity for introductory robotics – lots of information out there, but some of it is not very good…
- Appropriate background information will be provided, but now starting to move away from guided work in preference to posing problems to be solved

Session 3 Optical Line Following

Elements of a Line Following Robot

- **A robot**
  - ESP32 based and able to interface with other microprocessors and sensors
- **A method for detecting a line on the driving surface**
  - Choice of sensor
  - Sensor output – digital or analogue
  - Number of sensors
  - Interfacing to an Arduino
- **A control process to accurately follow the line**
  - Algorithms to stay on track

Session 3 Optical Line Following

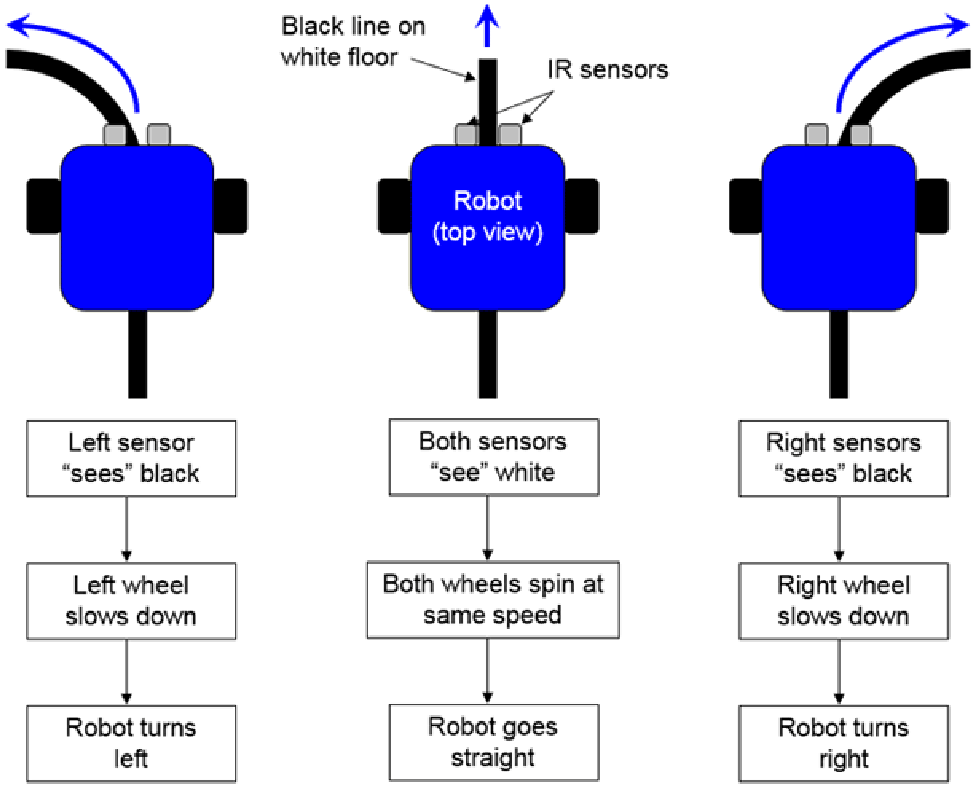
Lots of resources and Examples on the Web

- https://www.youtube.com/watch?v=3blywOpt8yo
- https://www.instructables.com/Line-Follower-Robot-Using-Arduino-2/
- https://www.electronicshub.org/arduino-line-follower-robot/

These are just a few from the 1st page of a Google search

Session 3 Optical Line Following

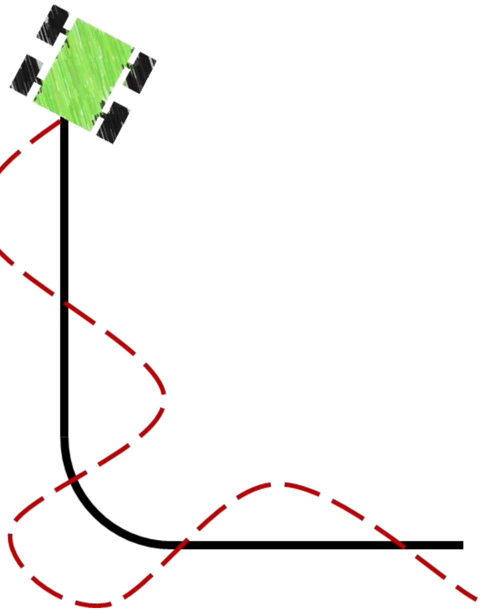
Simple Line Following Approach



http://www.sciencebuddies.org/science-fair-projects/project_ideas/Robotics_p023/robotics/line-following-robot.shtml

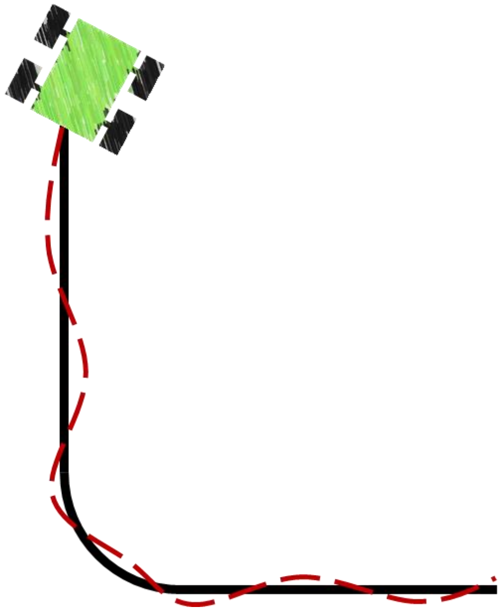
Session 3 Optical Line Following

Basic Line Follower



Vehicle follows line but with large errors

Advanced Line Follower



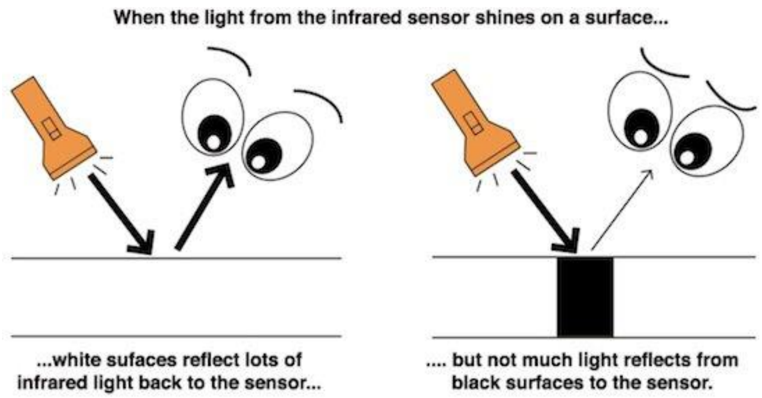
Vehicle smoothly follows line

Optical Line Following

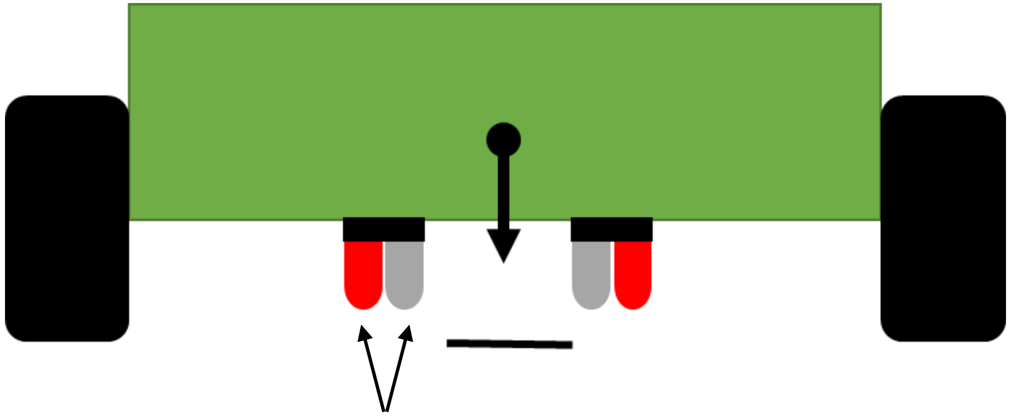
1. The Sensor System

Session 3 Optical Line Following

- We need a sensor system to detect the presence of a black line on a white background



Sensor signal is therefore **proportional** to the amount of IR light reflected from the surface



IR emitter/detector pair

Session 3 Optical Line Following

Two sensors connected to analogue inputs on Arduino

- Can provide basic information on distance from the line
- The number of times per second (**frequency**) we can interrogate (**sample**) the sensors for a value and use them is limited by the analogue to digital conversion (ADC) process implemented on the Arduino
- This ultimately limits the ability of the robot to follow a line at speed

Session 3 Optical Line Following

- **Two sensors can work**
- However, system requires careful calibration to relate sensor voltage to distance from the line
  - Calibration affected by
  - Line width changes
  - Distance from track
  - Ambient lighting
- If both sensors see black or both see white no information on which way to turn
  - Must implement a method to find the line which is time consuming

Session 3 Optical Line Following

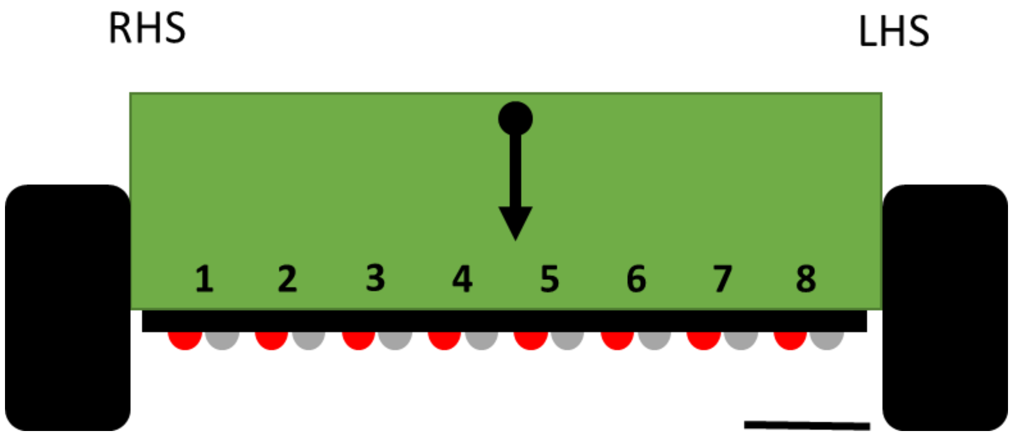
Sensors to be used

- We will use a number of IR emitter/detector pairs
- Connected to Arduino Analogue pins
- Pins accept input voltage range of 0 to 5V
- Converts voltage to a value between 0 and 1023
    - 10 bit Analogue to Digital Converter

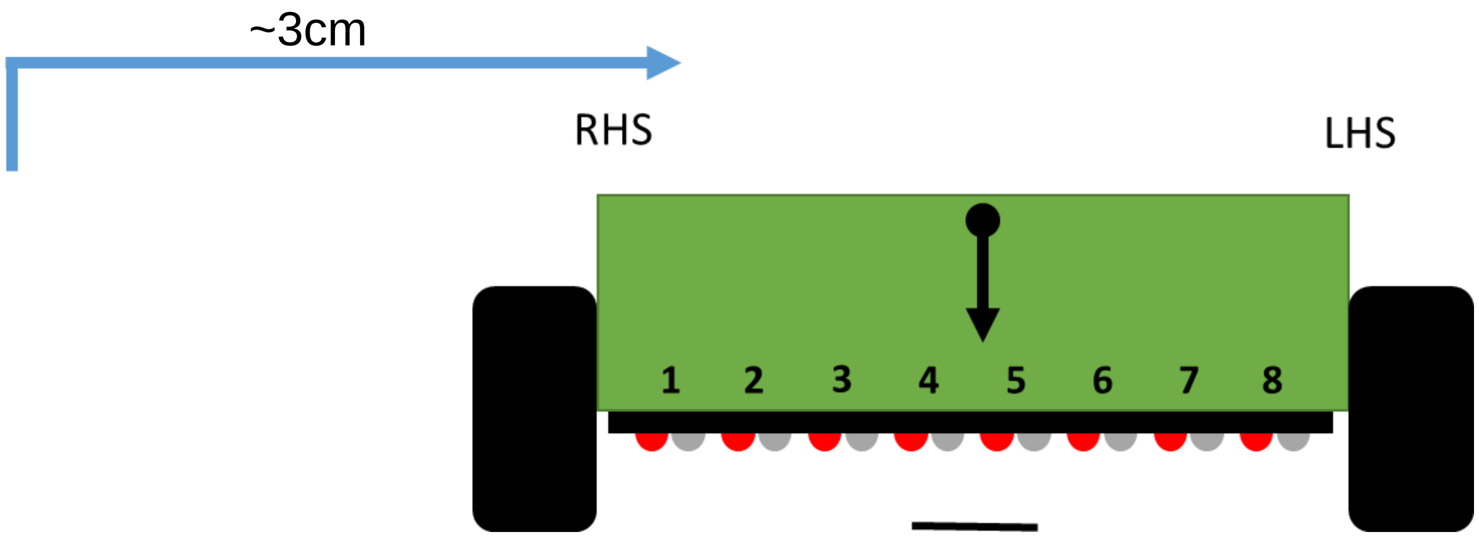
Optical Line Following

2. Measuring the Error – Deviation from the Line

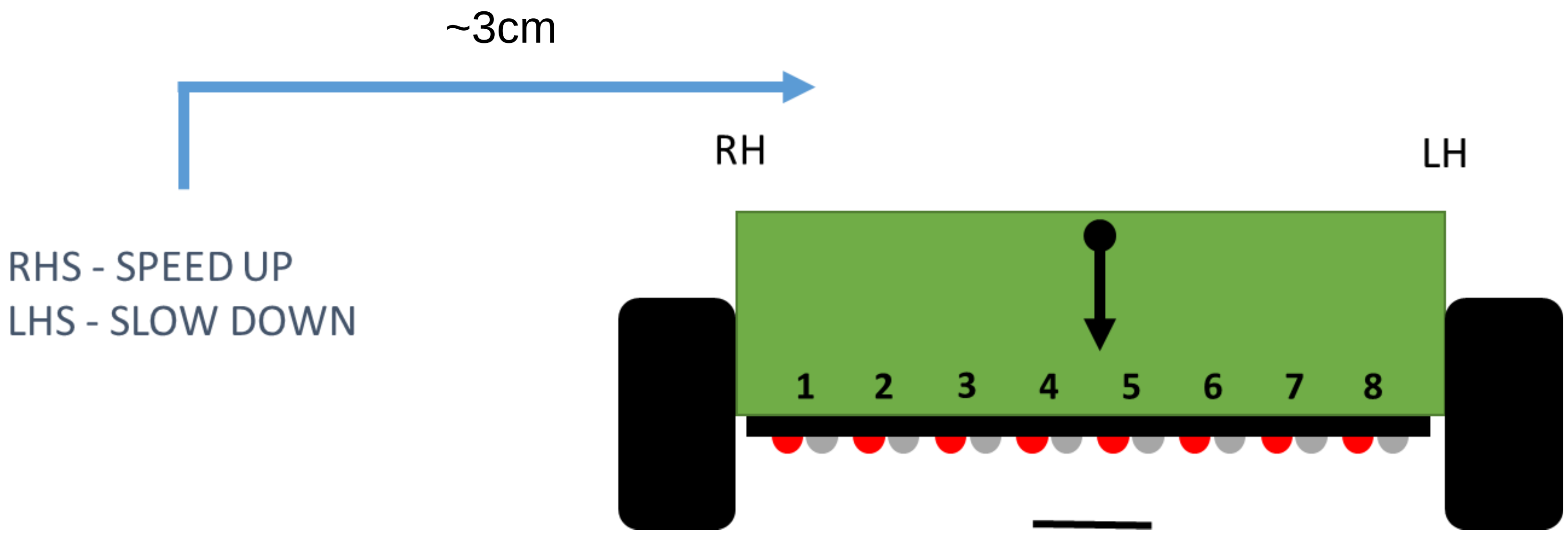
- **For speed and accuracy use multiple sensors**
- **Distance between sensors is known**
  - So for example: using 8 sensors, if the sensors are 1cm apart and sensor 8 sees the line the robots centre point is ~3cm to the right of the line, the robot needs to turn left to bring its centre line back over the line



- **For speed and accuracy use multiple sensors**
- **Distance between sensors is known**
  - So for example: using 8 sensors, if the sensors are 1cm apart and sensor 8 sees the line the robots centre point is ~3cm to the right of the line, the robot needs to turn left to bring its centre line back over the line

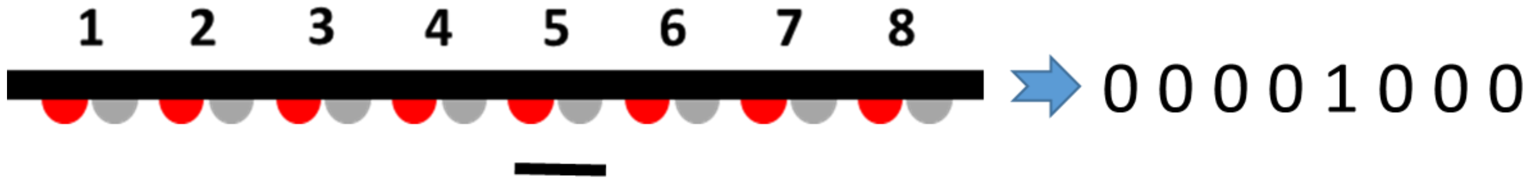


- **For speed and accuracy use multiple sensors**
- **Distance between sensors is known**
  - So for example: using 8 sensors, if the sensors are 1cm apart and sensor 8 sees the line the robots centre point is ~3cm to the right of the line, the robot needs to turn left to bring its centre line back over the line

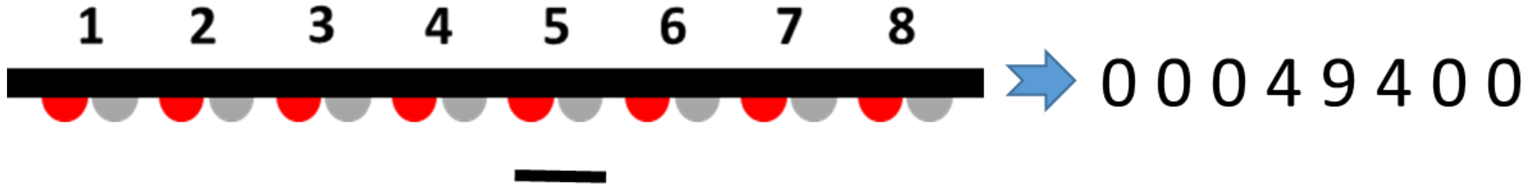


How do we read the sensors?

- Simple approach to obtain a binary on/off for each sensor (0=black, 1=white)



- A more elegant solution is to output a value that is proportional to the amount of the line each sensor sees e.g. black to white = 0 to 9



Weighted average in brief:

- Each sensor is calibrated to return a min and max value: e.g. 9 over black, 0 over white (max value always over line… implications??)
- Each sensor value is given a weighting: i.e. the distance from a reference point (e.g. here centre point of vehicle indicated by ●)

Sensor ID	1	2	3	4	5	6	7	8
Weight (mm)	-35	-25	-15	-5	5	15	25	35

- Distance (error to be minimised) from centre point is then given by:

$$Error = \frac{\sum_{i=1}^{8} sensor\ valuei.weighti}{\sum_{i=1}^{8} sensor\ valuei}$$

Weighted average in brief

$$Error = \frac{\sum_{i=1}^{8} sensor\ valuei.weighti}{\sum_{i=1}^{8} sensor\ valuei}$$

e.g. Only sensor 1 over black

$Error$ = [(-35*9) + (-25*0) + (-15*0) + (-5*0) + (5*0) + (15*0) + (25*0) + (35*0)] / 9 = **-35mm**

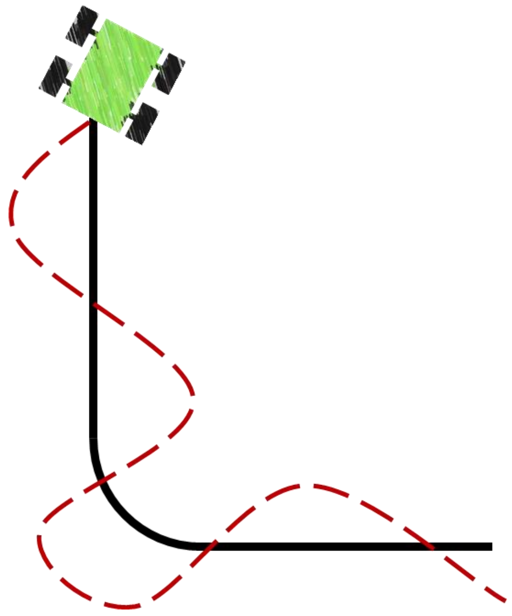
Sensor ID	1	2	3	4	5	6	7	8
Weight (mm)	-35	-25	-15	-5	5	15	25	35

Optical Line Following

3. PID Control

Session 3 Optical Line Following

- If we accurately know the magnitude of the error we can initiate a turn to bring the centre point of the vehicle back over the line
- But
  - How much do we turn?
  - How quickly do we turn?



Session 3 Optical Line Following

- If we accurately know the magnitude of the error we can implement a control scheme known as:

Proportional Integral Derivative (PID)

- PID properly implemented should enable the robot to follow the line in a stable manner

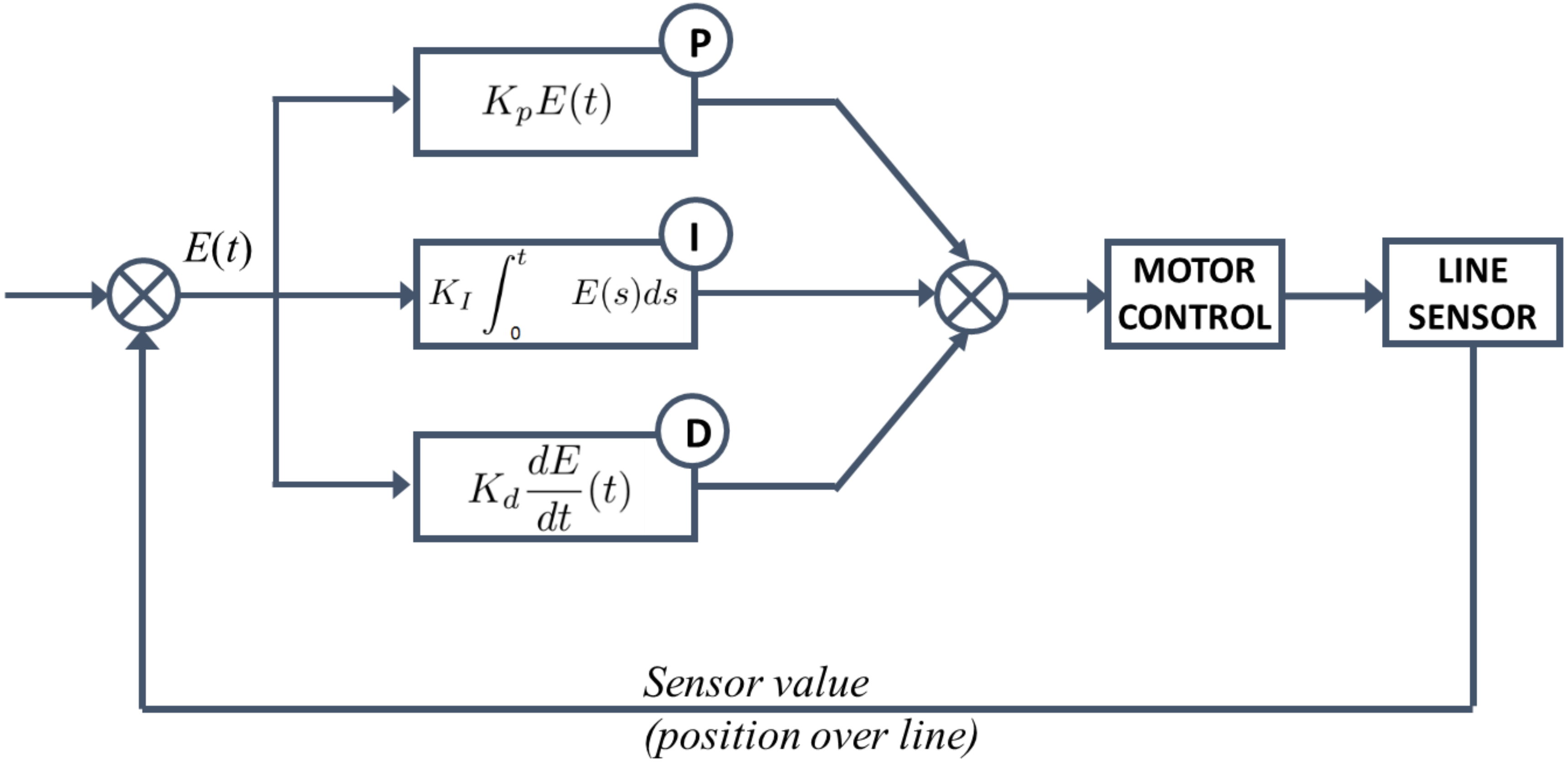
- **The theory behind PID control will be covered in specific control modules later in the course**
- **The formula will be provided**
  - You must implement it in code
  - The detail will be provided through flowcharts, comments or pseudo code
  - No new programming techniques are introduced
  - Knowledge acquired to date will be enough to complete the tasks to satisfy the objectives

Session 3 Optical Line Following

Key to a more accurate line follower using PID control:

- Measure the robot's centre point deviation from the line i.e. the **error** in our system
- If the robots deviates from the line return to it as quickly as possible
  - The amount of turn required to bring the robot back to the line quickly will be **Proportional** to the error
- Keep a log of the accumulated error over time
  - The **Integral** of the error
- But avoid overshooting the line as error will then begin to increase again and the robot will oscillate
  - Monitor the rate of change of the error, its **Derivative**

PID Process



P
$K_p E(t)$
I
$K_I \int_0^t E(s)ds$
D
$K_d \frac{dE}{dt}(t)$
$E(t)$
MOTOR CONTROL
LINE SENSOR
Sensor value
(position over line)

- **PID Control Theory is not required, however:**
- You will need to tune the controller to achieve the best performance
- You tune the controller by adjusting the $\boldsymbol{K_p}$, $\boldsymbol{K_i}$ and $\boldsymbol{K_d}$ values of the PID controller in the code provided
- The optimum values will be robot specific and specific to the track
- The process will be one of trial and error, but an approach will be suggested…
- **Tuning the PID controller K values requires changing the code**

PID Tuning Approach

- For this application, mostly trial and error, but
- Set K_p, K_i and K_d to zero
- Start tuning K_p only to try and get robot to follow the line [this is a P controller]
- Now tune K_d to increase straight line speed [this is a PD controller]
- Then tune K_i to improve performance as much as possible [this is now a PID controller]
- If it all goes wrong start again…

Session 3 Optical Line Following

Where to start….

- In order to successfully complete the required tasks, please familiarise yourself with all the documents on Moodle and only then begin work. Please take your time and don't rush. This is not a competition.
- There are no prizes for finishing first, whilst not understanding what you are doing
- The best coursework submissions will be based on acquired knowledge and not necessarily number of tasks completed…

Session 3 Optical Line Following

Session Plan: Groups A and B

- Week 1: In-lab
  - Work through all documentation
  - Design and build an n element sensor system
  - Implement a line follower using proportional control
- Week 2: Self study
  - Background research on optical sensors
  - Background research on PID control as applied to two wheel driven vehicles
  - Implement and tune full PID
- Following the session: In person demo of line following

Session 3 Optical Line Following

Session Plan: Groups C and D

- Week 1: Self study
  - Work through all documentation
  - Background research on optical sensors
  - Design n element sensor circuit
  - Background research on PID control as applied to two wheel driven vehicles
- Week 2: In-lab
  - Build an n element sensor system
  - Implement a line follower using PID control
- Following the session: In person demo of line following

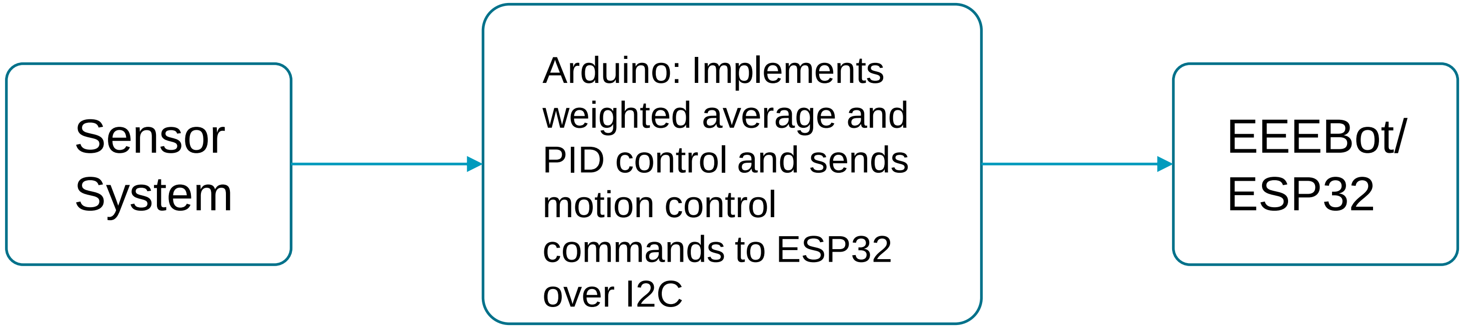
Session 3 Optical Line Following

Important

- You will need to be able to implement and tune PID for session 5 and 6.
- Assessment will be based in-part on technical progress. As an example of grade achievable for session 3 objectives (final detail will follow session 4)
  - Max 50% - Weighted average line following
  - Max 70% - PID line following – code seen by staff
  - +15% - demonstrating smooth and fast line following
  - +15% - other criteria

Session 3 Optical Line Following

System Overview



- You will write your own line following code, using PID control, and program this onto the Arduino NANO
- The Arduino will then send a left and right motor speed to the ESP32 over I2C, identical to Session 2
- You will use the same code on the ESP32 as you did in Session 2 – find this code under Session 2 called EEEBot_ESP32_firmware

University of Nottingham
UK | CHINA | MALAYSIA

End of Session 3
Introduction