IDP First Report

Team: root g
Robots: e, π

IDP Group L102

January 27, 12021

Downing College

Name	Lab Group
Cameron Spiers	120
Haymandhra Pillai	120
Akash Gupta	124
Noah Gordon	124
Pengyu Zhang	125
Tommy Rochuseen	126

Assessor	Prof Tim W.
Tech Advisor	Caglar K.

1 Task

1.1 Task Brief

- Design a robot and control algorithm to pick up coloured blocks and deliver them home
- Ensure that the robot doesn't collide with the other one
- Simulate the robot in Webots

1.2 Our Approach

We have decomposed the problem into 4 critical areas and who will be working on them:

1. Mechanical - Cameron & Haymandhra
2. Electrical - Akash & Pengyu
3. Software - Noah & Tommy
4. Simulation - Everyone

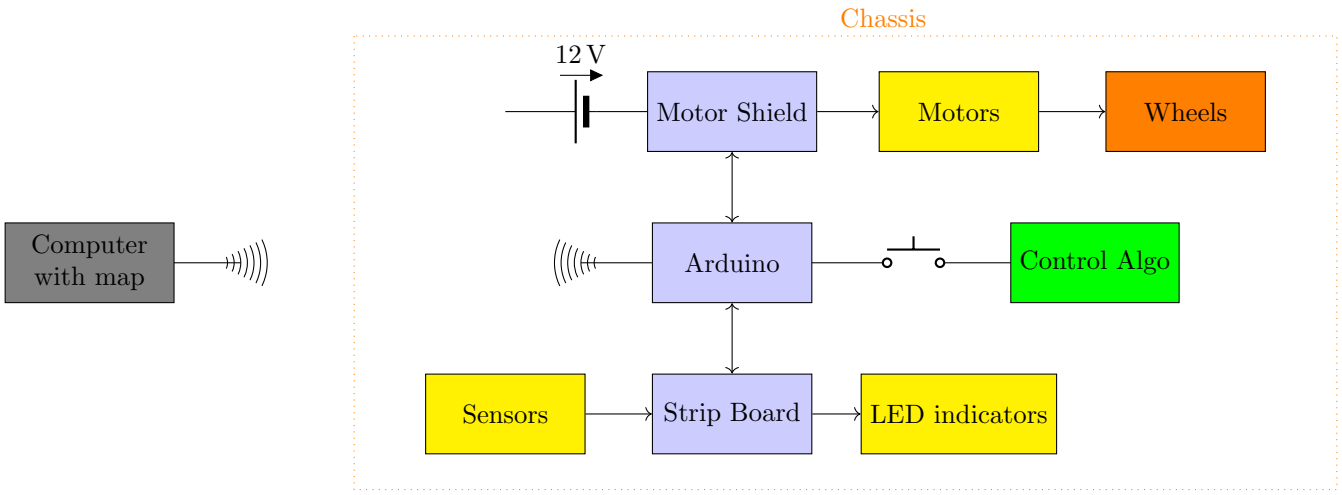
Our approach consists of tackling the first 3 parts in parallel before combining them together. However, we are aware that the biggest mistake is to not perform any integration testing, hence we plan to take an iterative approach, where we frequently test our ideas and models together in the simulated environment and go back to the drawing board to improve our design. Also, since the controller will be developed much later on, we have made a simple arrow key based simulation for the mechanical side to test their design.

1.3 Potential challenges

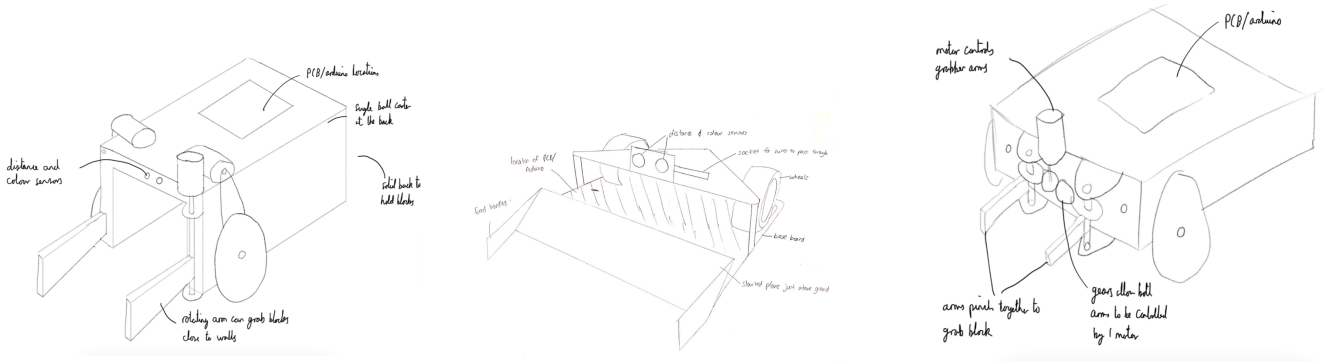
- Preventing collision with the other robot
- Ensuring the block is head on when picking it up
- Collecting blocks on the edges/corners
- Ensuring everything works together

We believe that best way to mitigate any delays in solving these problems will be to continually test what we have and rapidly prototype designs and controllers, which is reflected in our approach and Gannt chart.

2 System



3 Robot Concept

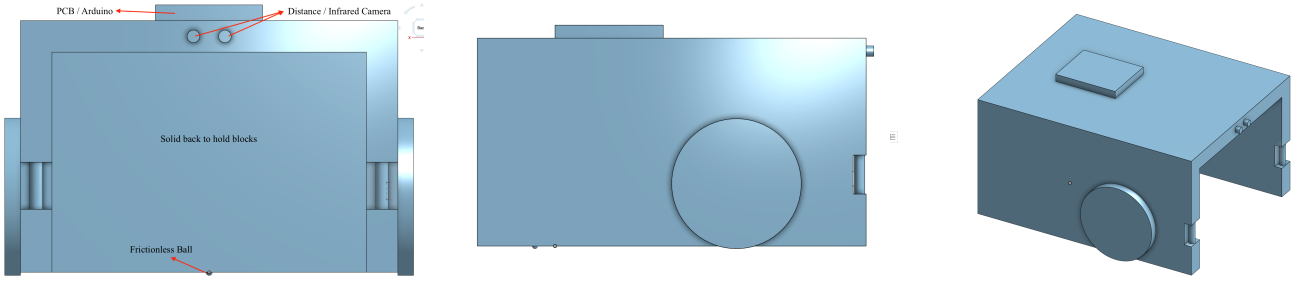


(a) Collector (b) Scraper (c) Picker

Figure 1: Initial sketches

Bot Name	Advantages	Disadvantages
Collector	Simply has to drive over blocks Can collect and transport multiple blocks	Handling is more difficult, and increased friction Complex pulley-belt mechanism for wheels
Scraper	Able to access blocks along corners and edges Lighter, hence a smaller motor Can guide multiple blocks	Blocks might not be safely guided to their destination Could slip while picking up blocks Wheel at the back due to the front bucket
Picker	Can easily grab blocks in any location The main body of the robot is very simple Less likely to slip	Complex gear and hing assembly to grab blocks Can only collect one block at a time Dragging blocks reduces handling

Table 1: Table evaluating the initial sketches



(a) Front (b) Side (c) Isometric

Figure 2: Simplified CAD model of Collector

Based on our evaluation of the initial sketches, we decided to design the Collector due to its simplicity and versatility. Once we test this design in the simulation, we will either improve this or pick one of our other designs. Also note that the design will be improved later on to consider the size of the electronic components.

4 Electronics and Sensing

Module	Purpose
Arduino	Micro controller - bridge between software and hardware
LEDs + Colour Filter + LDR	Colour Sensor
Ultrasonic Distance Sensor	Precise and high range distance measurement
Servo Motor	Control orientation of distance sensor for mapping
Hall Effect (Magnet) Sensor	Orientation Sensor
LED indicators	Indicate which part of the code is being executed
Motor Shield	Bridge between motors and Arduino
1308 Motor	High rpm motor to drive wheels

Table 2: Preliminary list of modules and their components.

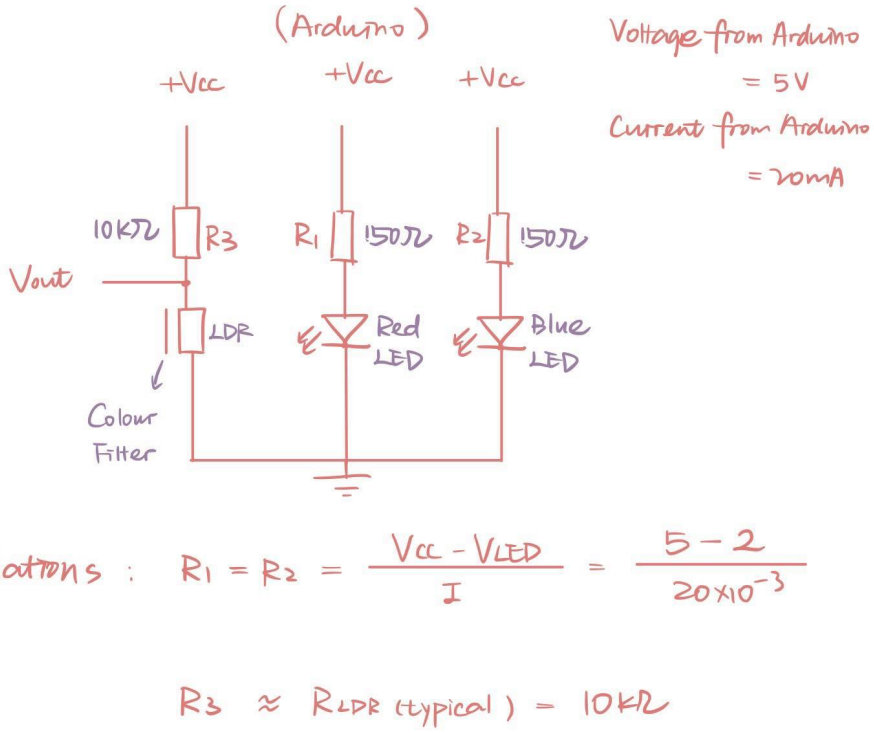


Figure 3: Schematic for colour sensor with calculations

The sensors produce an analogue signal, which will be read by the Arduino through an analogue input pin, which will be converted to a digital signal via an internal DAC. These will be fed into our controller to decide the next action. Notably, the range of output voltages for the sensors vary, hence this will need to be considered (e.g. through calibration) in the code.

5 Controller

This is arguably the most complicated part due to the constraints of simulation vs reality. The software will take inputs from sensors and use these to control the actuators and motors. LED indicators will be used for debugging purposes. We aim to mitigate the differences by designing and testing algorithms to cope for real life uncertainties and random errors. The algorithm is outlined as follows:

```
while not all the blocks have been collected do
    scan the arena for the other robot
    if the other robot is nearby then
        Move away
    end
    scan the arena for objects
    if a block is found then
        get closer to the block
        check the colour of the block
        if the block is the correct colour then
            pick it up
            if the block has been picked up successfully then
                navigate back home
            else
                try again
            end
            keep track of the blocks collected
        else
            mark this as not my block
        end
    end
    if a wall is found then
        turn right
    end
    Keep moving forward
end
```

Algorithm 1: Core algorithm of controller

There are simple pitfalls in this algorithm:

- How do we tell if the other robot is nearby
- How do we distinguish a wall from a block
- How do we tell if we're aligned with the block
- How do we tell if the block has been picked up
- How do we navigate the robot around

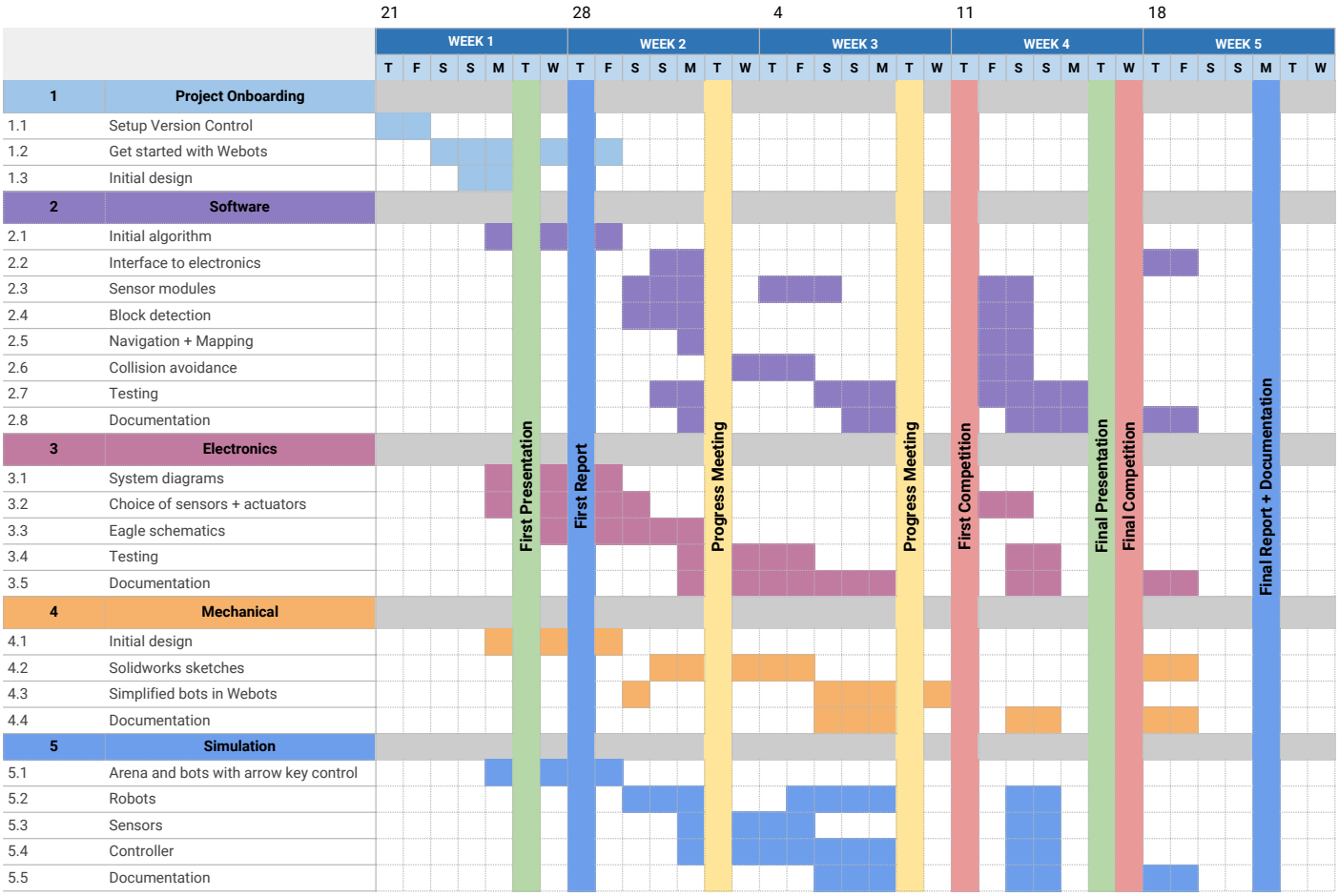
Some of these can be solved with software and some with the help of hardware. Algorithms for each of these questions will need to be implemented, but more detail can only be given after testing the pitfalls in the Webots simulation. Possible solutions are listed below:

- Construct a simple map of the block positions and robots, which can be updated by each robot
- Add a light sensor / push sensor to detect a block
- Navigation algorithms include A* or Dijkstra's, which require a map. Instead a simple snake search might be easier.

6 Timeline

IDP GANTT CHART

TITLE	IDP	**TEAM NAME**	root g
PM	Akash	**DATE**	26/01/21



1	**Project Onboarding**																																		
1.1	Setup Version Control																																		
1.2	Get started with Webots																																		
1.3	Initial design																																		
2	**Software**																																		
2.1	Initial algorithm																																		
2.2	Interface to electronics																																		
2.3	Sensor modules																																		
2.4	Block detection																																		
2.5	Navigation + Mapping																																		
2.6	Collision avoidance																																		
2.7	Testing																																		
2.8	Documentation																																		
3	**Electronics**																																		
3.1	System diagrams																																		
3.2	Choice of sensors + actuators																																		
3.3	Eagle schematics																																		
3.4	Testing																																		
3.5	Documentation																																		
4	**Mechanical**																																		
4.1	Initial design																																		
4.2	Solidworks sketches																																		
4.3	Simplified bots in Webots																																		
4.4	Documentation																																		
5	**Simulation**																																		
5.1	Arena and bots with arrow key control																																		
5.2	Robots																																		
5.3	Sensors																																		
5.4	Controller																																		
5.5	Documentation																																		

Figure 4: Gannt Chart for IDP Group L102