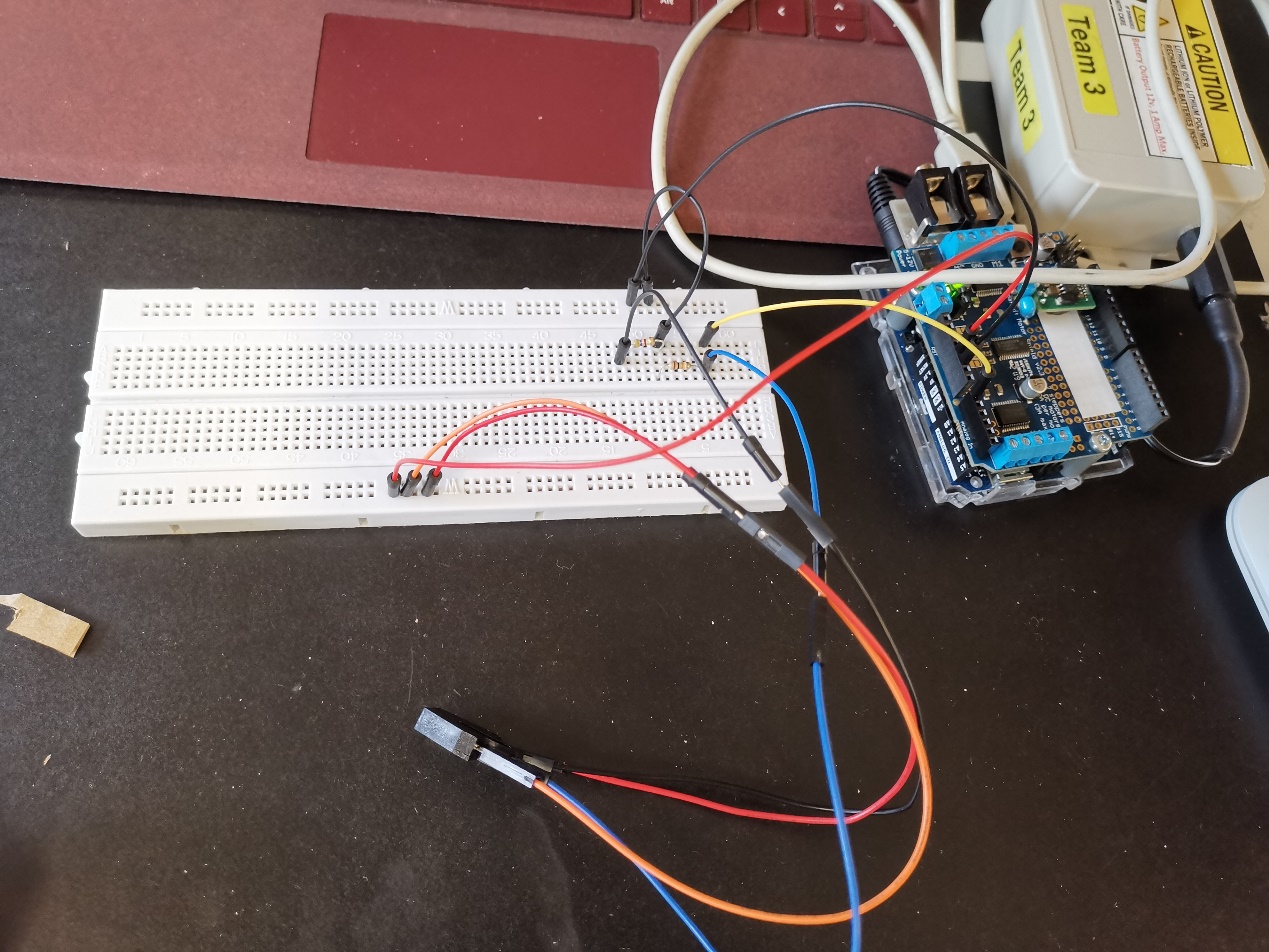
09/11

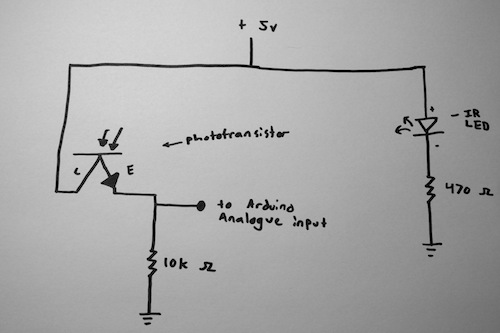
**Things done:**

**1. motor**. Simply connect motor to M1 or M2. Code: Motor\_test

**2. Line sensor**: refer to line\_sensor.jpg and use analog data reading code to read A0 data. our line sensor gives a maximum value of around 400-500, and can give < 10 when it's totally out of line (when partly in the line it gives values between these two ranges) (ps 1024 is the maximum value it can give and corresponds to 5 V)

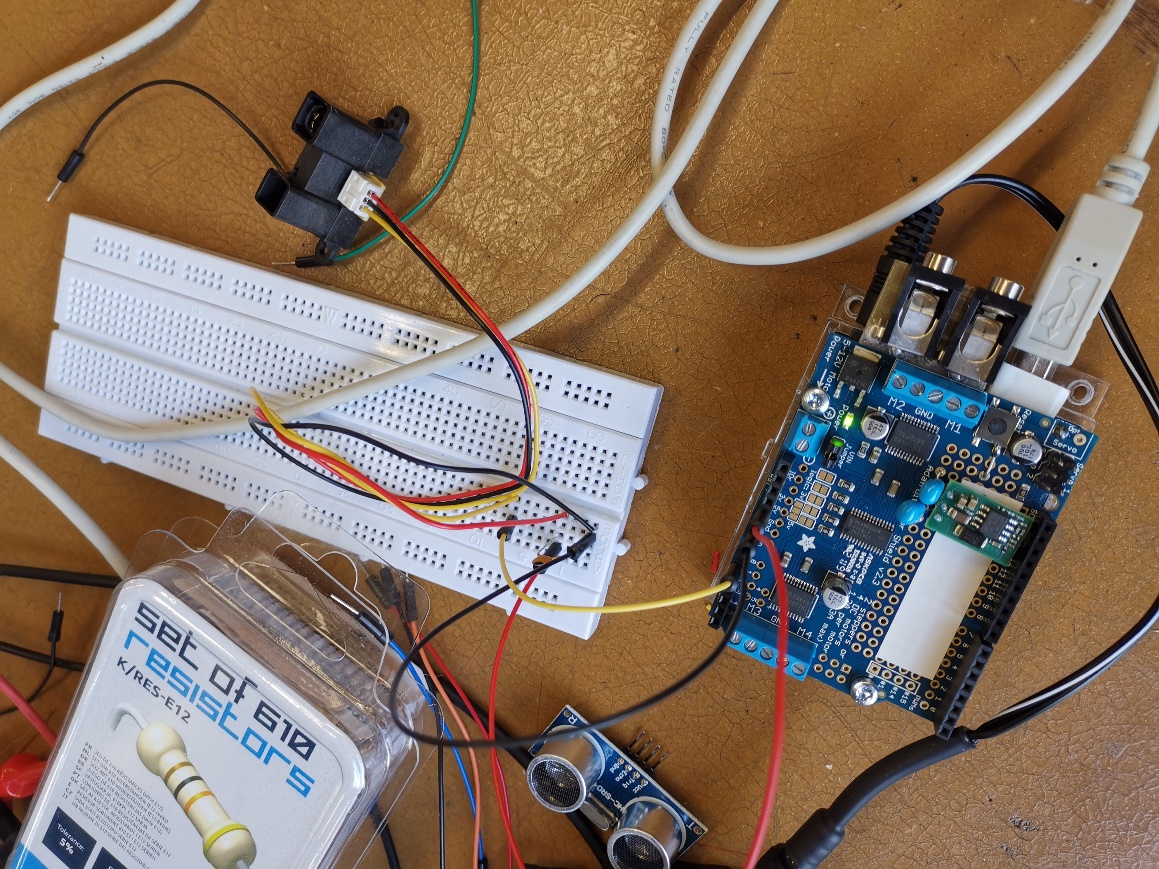


Line sensor



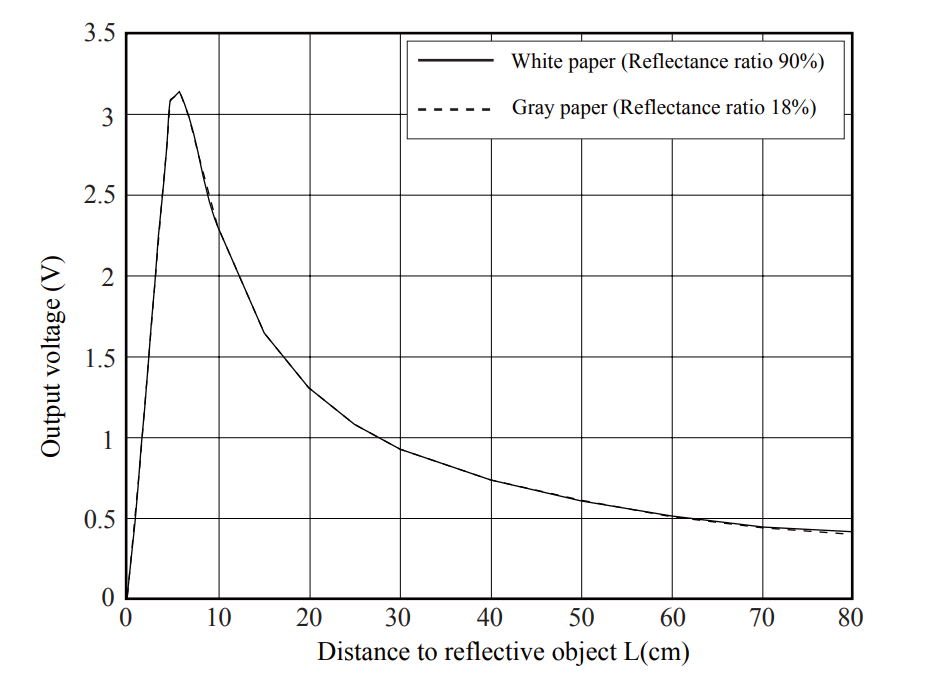
Line sensor circuit

**3. Distance sensor:** gives maximum value of around 600 at 10-20cm. circuit is simple, only add one 10 uF bypass capacitor between Vcc and GND (still quite unstable output but will test other capacitors to see if it works better)



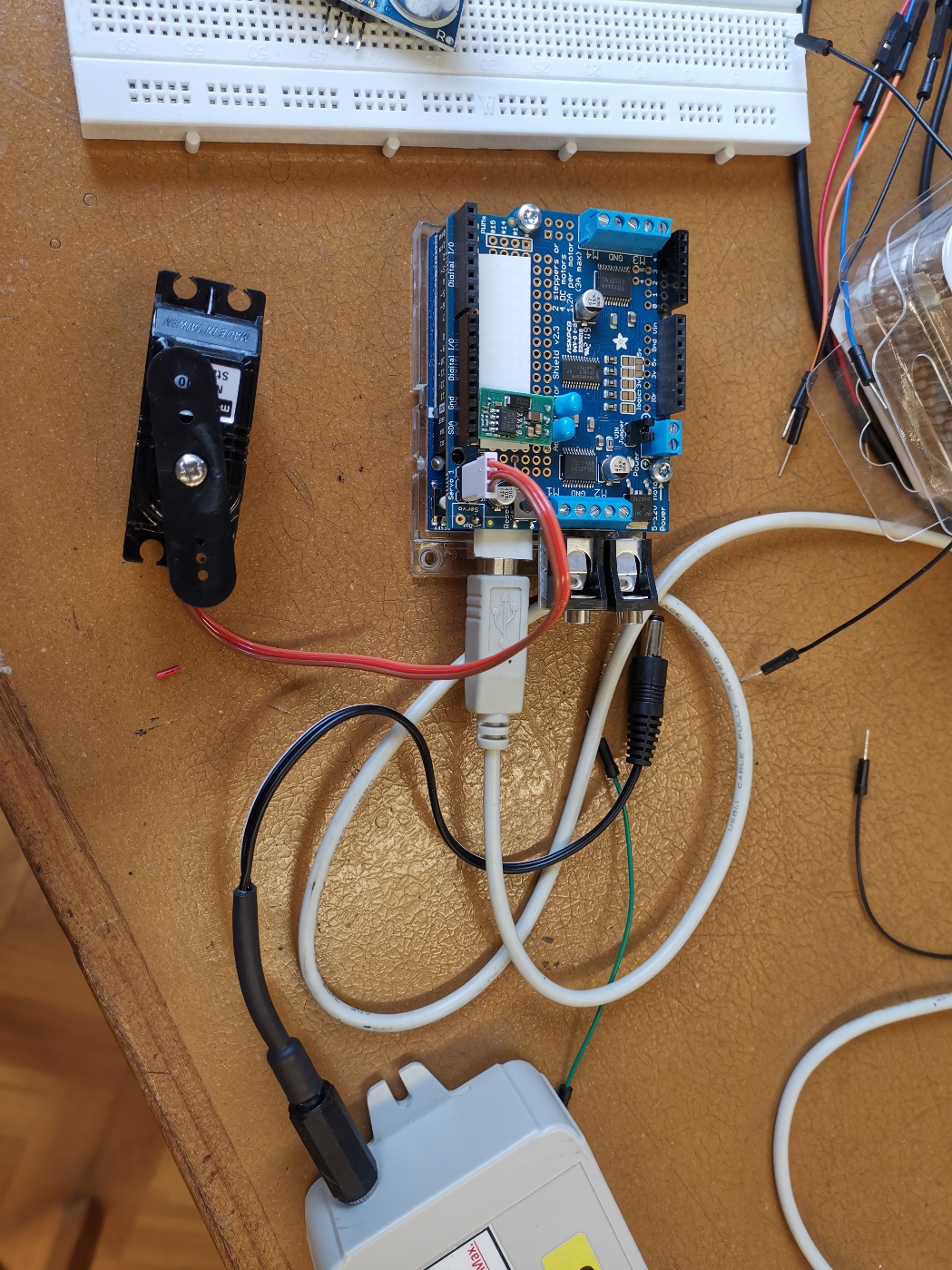
Distance sensor

Just for reference, this is the theoretical value:



**4. servo**

Code: servo\_test, pic: servo\_test.jpg



Servo\_test.jpg

Okay how is the line sensing going to work:

Line following algorithm:

* Keeps checks on which segment of line its on.+
* Base case is a straight line with robot on it – front two sensors register zero and back two above a threshold.
* Each sensor is checked once per loop and value stored. Increase in front sensor value is assumed proportional to deviation, and opposite side motor is increased in speed whilst corresponding side decreased (equal and opposite changes leading to no slip turning, somewhat like a differential. This allows for easy turning of corners.
* Upon moving along the line and thus through the tunnel, the robot will encounter a box ahead, as shown by its front mounted distance sensor. The robot then sticks on the line and tries to minimize this distance?
* The robot angles itself towards the box in question.
* The robot then, when the box is within grabbing distance, grabs the box by moving the servo to a specified grabbing angle and holding there, this should apply enough pressure on the box that it does not easily slip out / be slightly angled to enclose the box, duty of the mechanics team.
* Having got box, immediately analyses colour. Then retraces steps to return to the line
* Attempts to take box to first location ( known on which line segment prior) When hits line, turns and follows to the right direction to find the object (keeps track of inertial cords by. Colour sensor pointed downwards locates region.

REPORT WRITING stuff

* **Electronics/Sensing.** This should include a list of sensors/circuits required, any circuit diagrams/block diagrams which may have already been developed. Discussion as to if/why some processing will be performed in electronics opposed to software (e.g. obtaining digital outputs from analogue signals)
* **Software.** Exploration and navigation algorithms. Interface to electronics, discussion of choice of algorithms, any failure detection/recovery which will be implemented.
* Integration between hardware electronics and software

Integration between hardware and software.

Sensors used at present

* 2 \* analogue line sensor inputs
* 3 \* digital line sensor inputs
* 2 \* digital colour sensor inputs
* 2\* distance sensor inputs
* 3 \* LED’s (red, blue, flashing)
* 2\* DC motors
* Single Servo for picking up boxes

Using two analogue line sensors at the front as well as three digital output line sensors on each side + at back for junction recognition and line alignment help. Colour sensor input digital for red or blue, and second downwards facing colour sensor for target location recognition.

Interface to electronics via a number of IC’s -> analogue to digital for colour and line sensors to remove dependence on analogue Arduino pins.

Also using 1 or 2 distance sensors to measure the distance to the box but also to measure the distance between the robot and any wall if they happen to be there.

Line following currently tested using both a simple PID controller, varying the differential speed of the wheels in response to the difference between the outputs of the central two line sensors, and a binary sensor which increases or decreases the differential speed of the wheels in response to the triggering of the left or right sensor.

Junction detected by the triggering of one of the digital sensors off axis from the wheels. These are only triggered by a 90 degree turn, and also allow for accurate 180 degree and 90 degree turns.

Section of track determined by the output from the sensors, given essentially by a Markov diagram, helps with faulty box detection / line following.

Code modular, with a central class Robot with all variables and function such as line following, each returning void and so can be run independently. This means that each part of the movement can be improved independently, and a central decision algorithm can decide which action to run at any given point.

The central algorithm has the goal of collecting finding and placing all boxes. It works by attempting to execute objectives one after the other. If the number of remaining boxes is greater than zero, it looks for a box, then classifies it as red or blue and looks for a suitable location to put it in, with priority to the least accessible placing locations. Then it returns for another box.

Two methods for avoiding box obstructions -temporary placement of boxes in a position or the moving of the robot around any obstructions that might arise on the line. The second of these could be simpler, but may present harder line following problems.

Failure detection and recovery

Should the robot leave the line, the back middle sensor will register the change as it leaves. This only happens during 90 degree and 180 degree turns, which have their own separate algorithms. So if this occurs outside of one of those algorithms then the robot is alerted, and will turn in the direction of the last triggered frontal sensor to regain the line. Should the robot leave the line to pick up a parcel, it regains the line by using any of the line sensors to detect a line, then using a combination of its orientation, which it keeps track of, and the line sensor that was triggered in order to perform the correct recovery response.

Risky Aspects

Leaving the line for any reason e.g obstruction avoidance