CPSC 1000: Introduction to Computer Science – PART 2

The project challenges

Robert Benkoczi, C556
robert.benkoczi@uleth.ca

25-oct-2018
(Week 7)

Objectives

- CPSC 1000 course project: students will program a three wheel robot controlled with an Arduino microcontroller that will seek a target object.
- Resources:
  - Arduino Notebook
  - Arduino Programming Language Reference
  - The Adafruit motor shield documentation: `https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/library-reference`
  - The course notes (the probing/action programming template, working with various sensors, etc.)

Introduction

The robot:

- Two motors controlling one rubber wheel each.
- To move forward, choose the “same” speed to the two motors.
- To turn, set different speeds to the two motors.
- Various sensors can be attached to the body of the robot.

Three challenges (target seeking robot)

The project is decomposed into three challenges, increasing in complexity.

- Task for all challenges: the robot should approach the target object (a box) within 20 cm from it, but without touching it.

- Setup for Challenge 1: robot facing the target object at distance 1 m from it.
- Setup for Challenge 2: robot facing the target object at an arbitrary distance between 1 m and 2 m.
- Setup for Challenge 3: set at an arbitrary distance between 1 m and 2 m from the target, oriented arbitrarily.

- The target object: box with a source of light on top.

Grading: 1 mark for each challenge + project log.

Project log

→ use exercise book, or sheets of paper (in binder), or a .doc file.

Content:

- date:

  → the task. Ex: "challenge 1, robot moves on a straight line".

  → the strategy to fullfill the task (your idea). Ex: "program robot to move for 1 sec & adjust the speed of left & right motors"

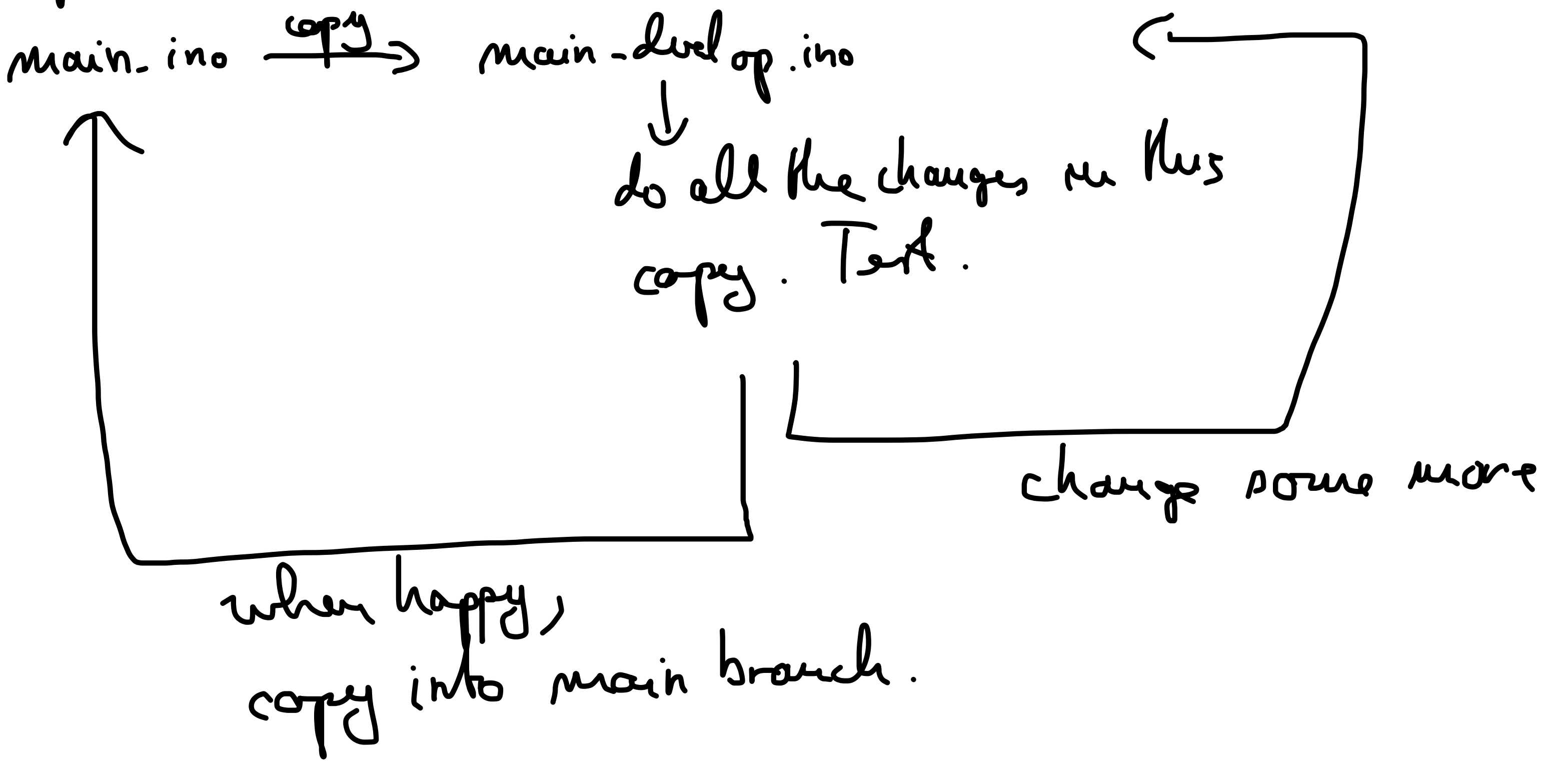
  → observations / conclusions: ex" speed chosen: x for left, y for right. If robot moves for 3 sec., trajectory off from straight line by only 5cm".

Advice :

- -> backup your code frequently.
  - -> Have the following source code files (branches)
    - -> Main branch : contains the code for your project.
      - -> ex : main. ino
    - -> Development branches, to try ideas.
      - -> distance. ino : develop a function that correctly reads the distance to an obstacle from the distance sensor.

Copy functions from development branches into the main branch, only when you are satisfied that the code copied works.

→ Make copies of the main branch



do all the changes in this copy. Test.

change some more

when happy, copy into main branch.

Challenge 1:

- → straight line trajectory
- → move forward for 80 ... 100 cm. (no sensor)
  ⊛

Problem: move robot → straight line
- → use delays until the robot starts moving
- → move robot using same speed for both motors.

- → move for 90 cm.
- → incorporate an on/off button ?

<u>Ideas</u> → what physical speed, in m/s corresponds to the chosen speed values that move robot on a straight line.

Experiment: run robot on a straight line for 1 sec. Measure the distance travelled. The measured value = speed, $v\left(\frac{m}{s}\right)$

$$v = \frac{dist}{time}.$$

Move 90 cm: use a delay $t = \frac{dist}{v} = \frac{0.9}{v}$ (Challenge nr 1)

Challenge 2

-> use the sensor to measure distance to an obstacle.

Overall design, probe / action

good candidate for state of system.

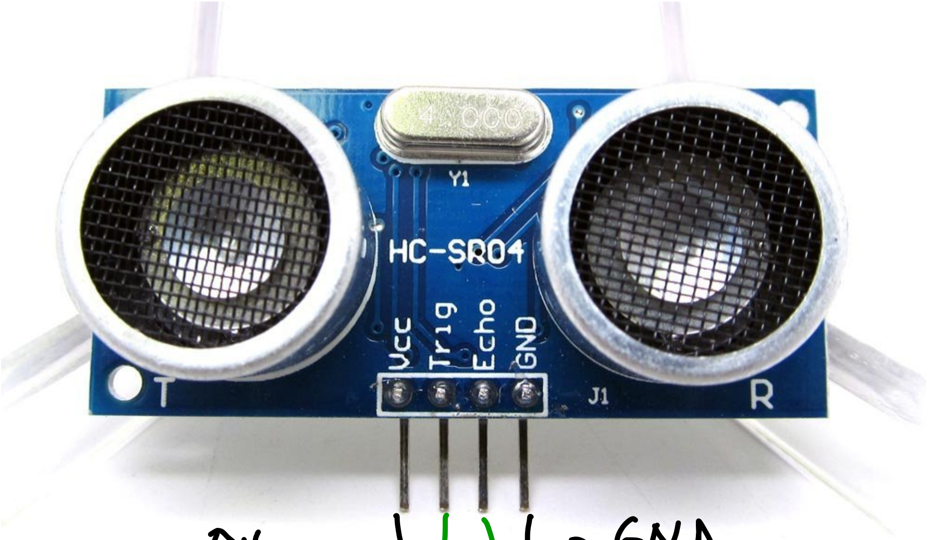
void loop() {

probing { - store distance in a $\boxed{\text{variable}}$

action { - move forward if distance > 20 cm otherwise stop.

}

Ultrasonic distance sensor - HC-SR04



+5V on Arduino

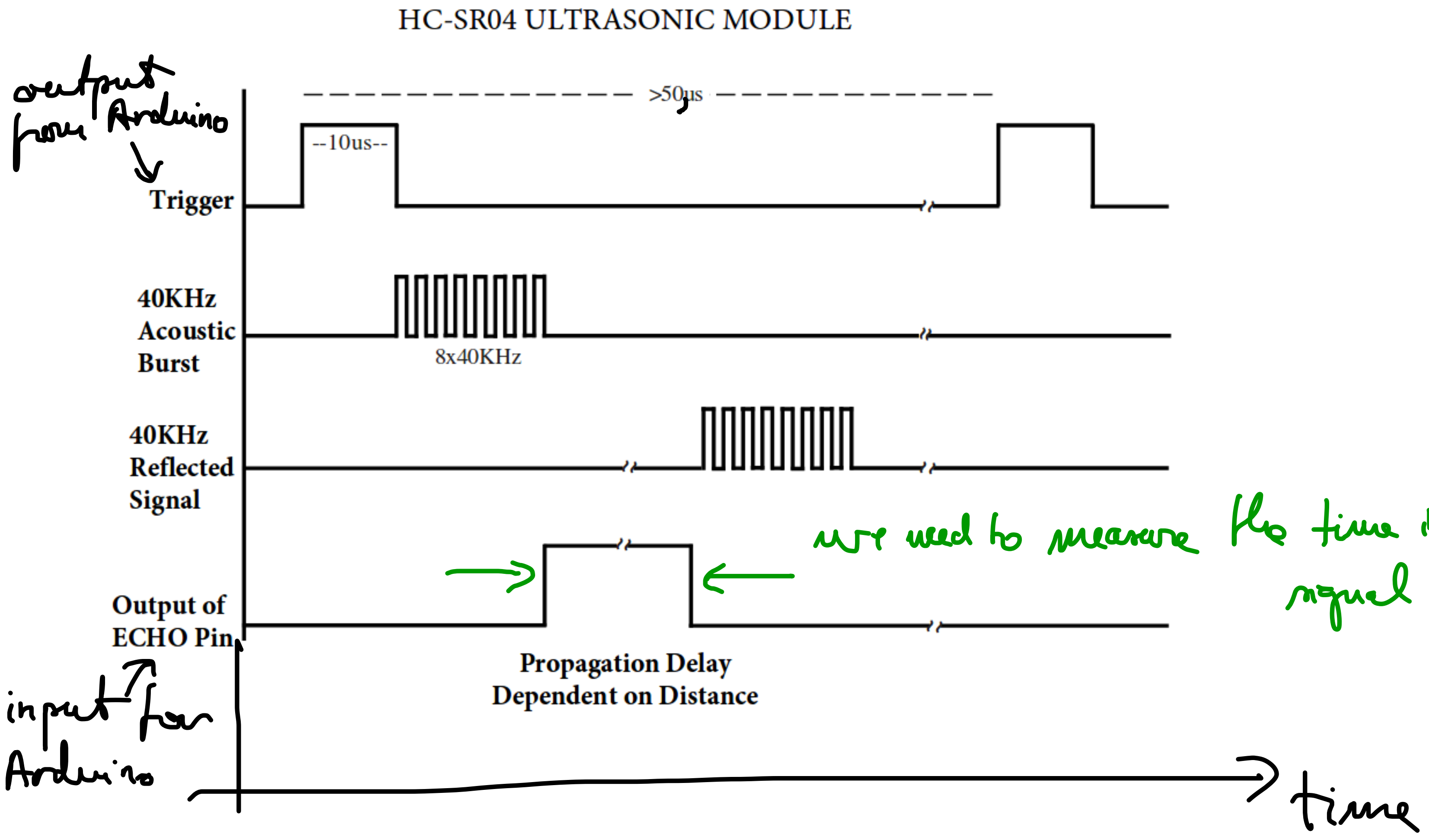
GND on Arduino

https://www.mpja.com/download/hc-sr04_ultrasonic_module_user_guidejohn.pdf

trigger (Arduino outputs a digital signal)

Echo (Arduino reads the digital signal)

Operation and timing (from manual)



we need to measure the time interval this signal is HIGH.

Arduino program

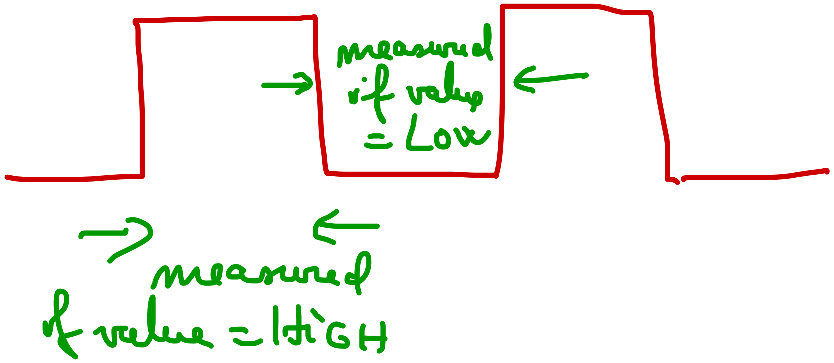
`pulseIn(pin, value)`: measures the duration of a pulse (μs).

see Arduino reference

→ return value = duration in μs (10^{-6} sec)

pin : integer identifying the digital pin from where we read the signal

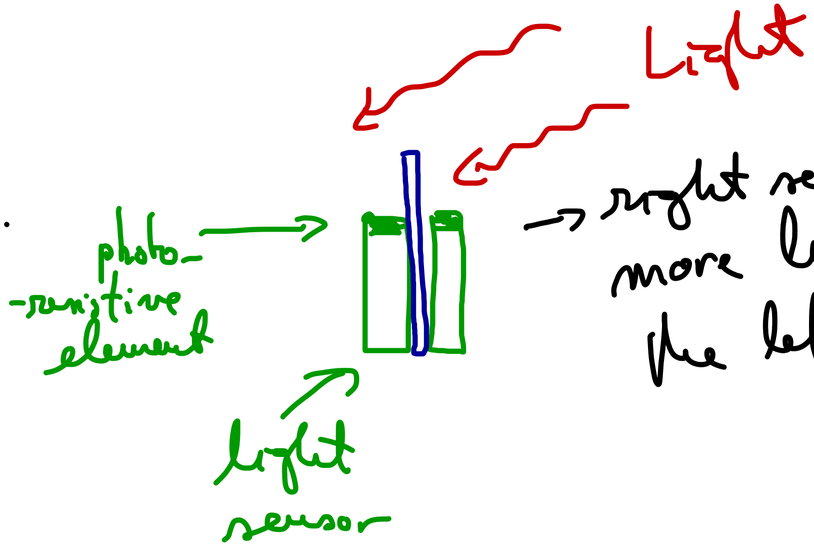
value : which part of the signal we measure → HIGH → LOW



Using this duration + speed of sound @ 20°C, we can estimate the distance to the obstacle.

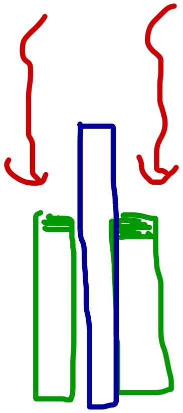
Challenge #3

We can use 2 light sensors.



→ right sensor measures more light than the left sensor.

See lab assignment 2 (potentiometer) to read the light sensors.



→ both sensors should detect the "same" intensity of light

Some "calibration" will be needed.