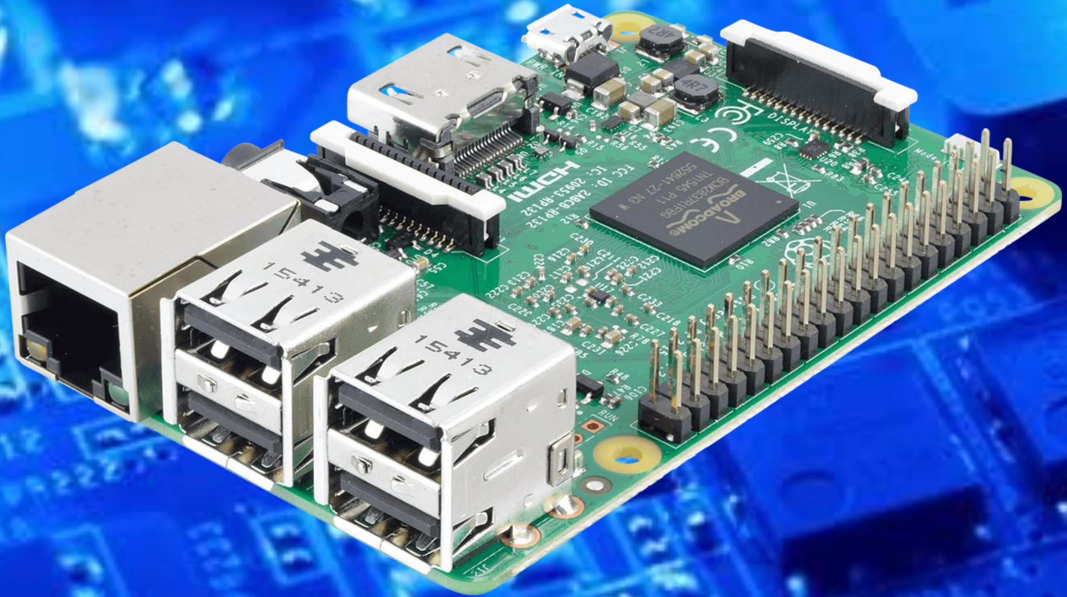
EEE3096S



P06 Miniproject & CS ES Project

This content it's examinable

P6

Embedded Systems II

Dr Simon Winberg

Electrical Engineering
University of Cape Tow

Outline

- About Prac6 / Mini-project A
- CS Mini-project B

Prac6 / Mini-Project A

(All students to complete)

This can be done as a team of 2 students

Prac6 / Mini-project – work schedule

EEE3096/5S

Mission:

'Twiddle Lock'

Practical 6 / Project A

&

ELO5.2 Assessment (attempt 1)

Prac06 is planned more around being a miniproject, the schedule is as follows:

25 September :
Prac06 commences (but you can finish off Prac05 still on this day)
05 October :
Prac05 due
12 October :
Recommended date to submit Prac06 (could extended to 19 Oct)

Prac6 / Mini-project – purpose

Prac6 is about developing a 'representative embedded system'

So it needs to have characteristics of being:

> "A task-specific computer, which is built into a larger system for the purpose of controlling and monitoring the larger system." [Catsoulis,2009]

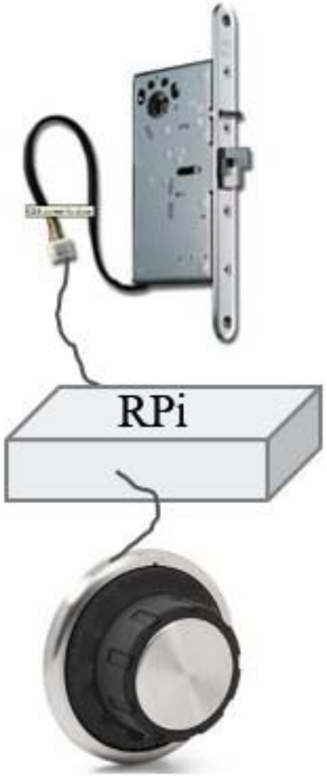
Which implies at least some of these:

- Specialized control & processing software (HW in the loop)
- H/W interfacing
- Real-time timing/operational constraints
- (We may need to improvise/imagine the built-in aspect)

Prac6 & ELO5.2 Testing

- *ELO5.2 stipulates:*
  “Creates computer applications as required by the discipline”
- *More specifically for a embedded computer application:*
  In Practical Assignment 6 the student is tasked with designing a small embedded system application that will use a digital-to-analogue convertor to sample a signal and implement specified signal processing operations guided by specification models (specifically use case diagrams and state charts)

Prac6 – The Mission



You are tasked to develop the ‘Twiddle Lock’.

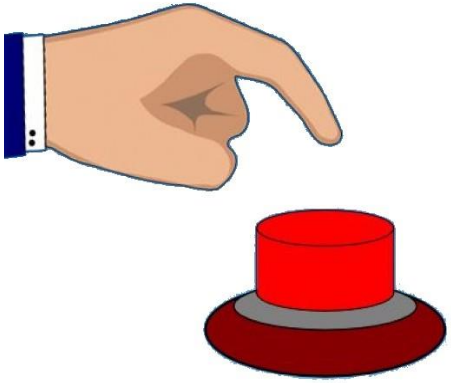
- This is an electronic form of the lock for a the classic ‘dial combination safe’ (DCS) combinational lock mechanism.
- But you will use a Raspberry-based application to emulate this behaviour
- The user will ‘dial in’ a combination code, using a potentiometer (POT) sensed by an ADC
- When the right combination code is entered the device will either
  - Send a lock signal (over the L-line) if the lock is currently unlocked and make an ‘locking sound’, or
  - Send a unlock signal (over the U-line) and make a ‘unlocking sound’ if it is currently locked.

How the combination code is entered…

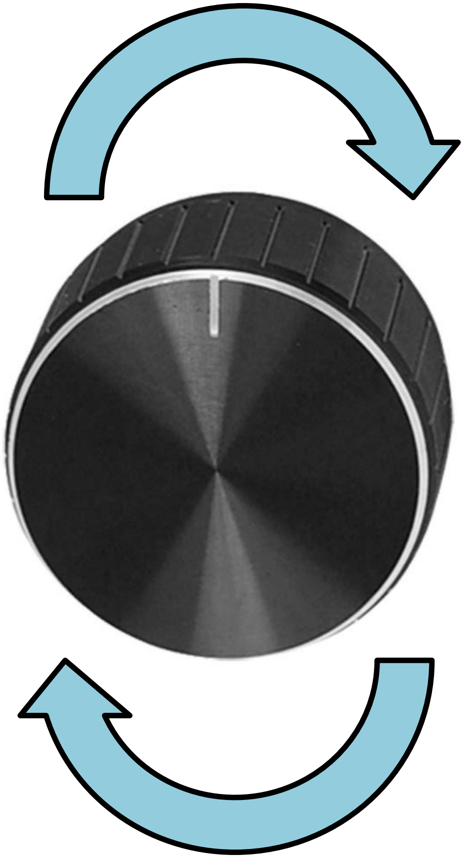
Prac6 Combination Code Entering

The user interface is pretty much like you might expect for a dial combination safe…

You enter the code by:



Press the S (service) Button for a short time



Turn in one direction for a certain duration

Repeat twiddling the knob a few times and then leave it for >2s and it will authenticate the combination and lock/unlock if valid

Turn in the other direction for a certain duration

Coincidentally note there aren't necessarily any markings on the knob! It is time and direction dependent.

Prac6 Combination Code

- The combination code is essentially a sequence of rotating the knob to the left or to the right.
- A combination code can be represented as a sequence of symbols
- The symbols were are using here can be represented as
  - Direction of the turn: Left(L) of Right(R)
  - Duration of the turn (time x100ms)

e.g. a code can be expressed in the form “L1R5” to mean turn left for 100ms and then turn right for 500ms

Prac6 Combination Code

- Note that the system should allow for either
  - A short delay (<1s) of not moving the dial to mark the end of a symbol   OR
  - Just turning the dial in the opposite direction to signal the end of a symbol that that you are immediately starting a new symbol
- If you leave the dial for longer than 2s this just mean ‘enter’ that you are done entering the code

Prac6 Lock/Unlock Sound

- Since there isn't really anything to lock or unlock your device is required to…

Produce a happy sound on successfully unlocking

(these could also just be short/long buzzes)

Produce a sad sound if you entered the wrong code

Produce a locking sound if you've successfully locked the device

Royalty Free Sounds from http://soundbible.com/

Parts and Software to Develop

- Parts
  - These are as per Prac4 but you might want to get a nicer knob and POT
- Software
  - This is explained further in the assignment description
  - One significant aspect is a ARM assembly code *sort* function

ARM Assembly Function

- You are to develop a function
  sort(int* x, int* y, int n)
- Which will sort values in *x* into order from minimum to maximum, stored into *y*
- We are not suggesting using in-place sorting because (e.g. Bubble Sort) is a bit more difficult to implement in assembly
- You are suggested to use selection sort because this is fairly easy… you can modify array *x* by e.g. by replacing values that have been moved into *y* with a negative number

ELO Testing

- ELO 5.2 tests need to be scheduled with a tutor, TA or lecturer once you have completed the project
- It is important for both students to work on the project!
- Both students need to be prepared to answer specific questions to confirm they have passed the ELO requirements

CS ES Mini-Project B

(only CS students to complete)

This can be done as a team of 2 students

The Mission – WaveSensor

- The plan for this project is to develop a type of IoT device, namely:
  - The "WaveSensor"

Basically the device has a light sensor that reacts when you wave your hand between the light source and the light sensor at a certain rate.

The wave needs to be done between the light source and the sensor so that a shadow is cast across the sensor which it will be able to detect.

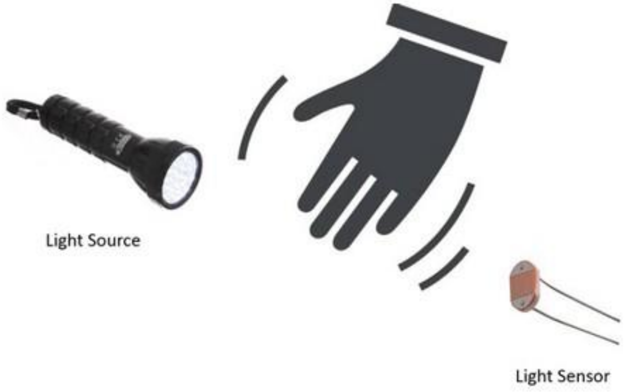


Figure 1: IoT WaveSenor

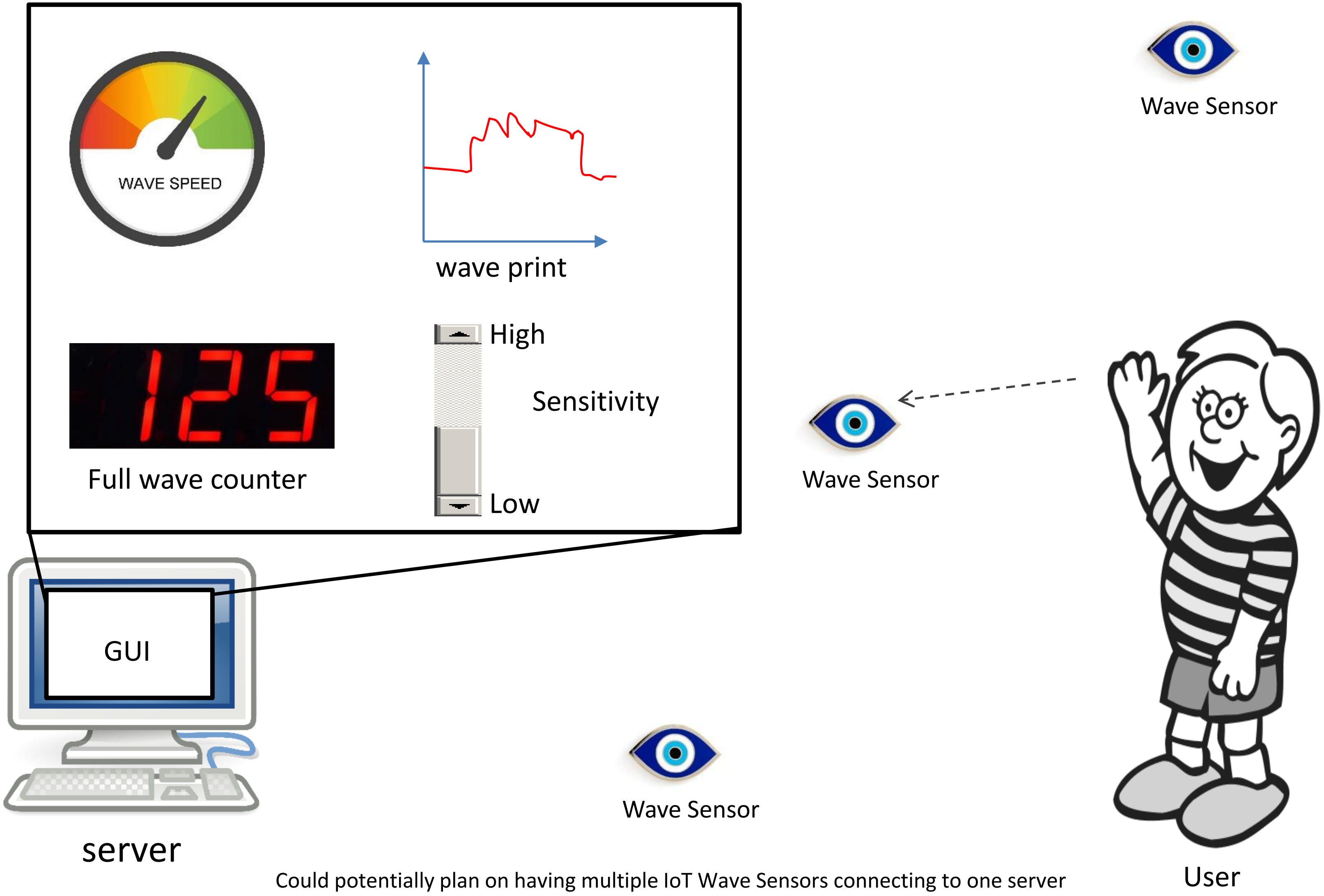
You need to try and detect the speed of the wave… this may be done in two ways:

- The rapid fluctuations in the shadows caused by fingers (advanced option) or
- Just the more significant changes in light as the whole hand shadow makes the shadow appear and disappear

The system

- This needs to be a type of IoT device, so
  - Waving information needs to be sent to a central server (which can be implemented as a simple Python / Java or C socket program)
  - There needs to be configuration settings that can be similarly be sent via the socket connection to the device to e.g. turn it on/off and change the sensitivity
  - Ideally (considering you all are great programmes) you should include a GUI on the PC (server) side to provide an interesting display (e.g. a counter for number of waves and a speed indicator for the rate of waving)

WaveSensor Concept Diagram



wave print

Full wave counter

High

Sensitivity

Low

GUI

server

Wave Sensor

Wave Sensor

Wave Sensor

User

Could potentially plan on having multiple IoT Wave Sensors connecting to one server

Any Questions??