Lancaster University

SCC.369: Mini Project

Bass Guitar

Mark McElroy m.mcelroy@lancaster.ac.uk
Harry Almond h.almond@lancaster.ac.uk
Christopher Sixsmith c.sixsmith@lancaster.ac.uk
Alex Carolan a.carolan@lancaster.ac.uk

Tasks

- **Planning**. Deciding upon what everyone needs to do, and how the device will work.
- **Building The Guitar.** Obtaining hardware. Assembling the cardboard body for the guitar, wiring up individual fret plates, and sorting out the internal wiring.
- **Fret Chips/Communications.** Programming the PIC mikro chips (in C) that control each string, allowing them to receive capacitive touch inputs and return a value over I2C declaring which of the 4-8 controlled frets are pressed. (5x chips needed, all running the same program with a different I2C address)
- **Synthesiser Module**. Programming the AdaFruit CPX module that polls the fret chips for the current pressed values over I2C and is capable of producing twenty four (24) different pre-synthesised bass guitar sounds via audio output.
- **Testing and Tweaking the Guitar –** ensuring final guitar is suitable for demonstration
- **Presentation**. Making the presentation to show to the class at the end of the task.

Resources

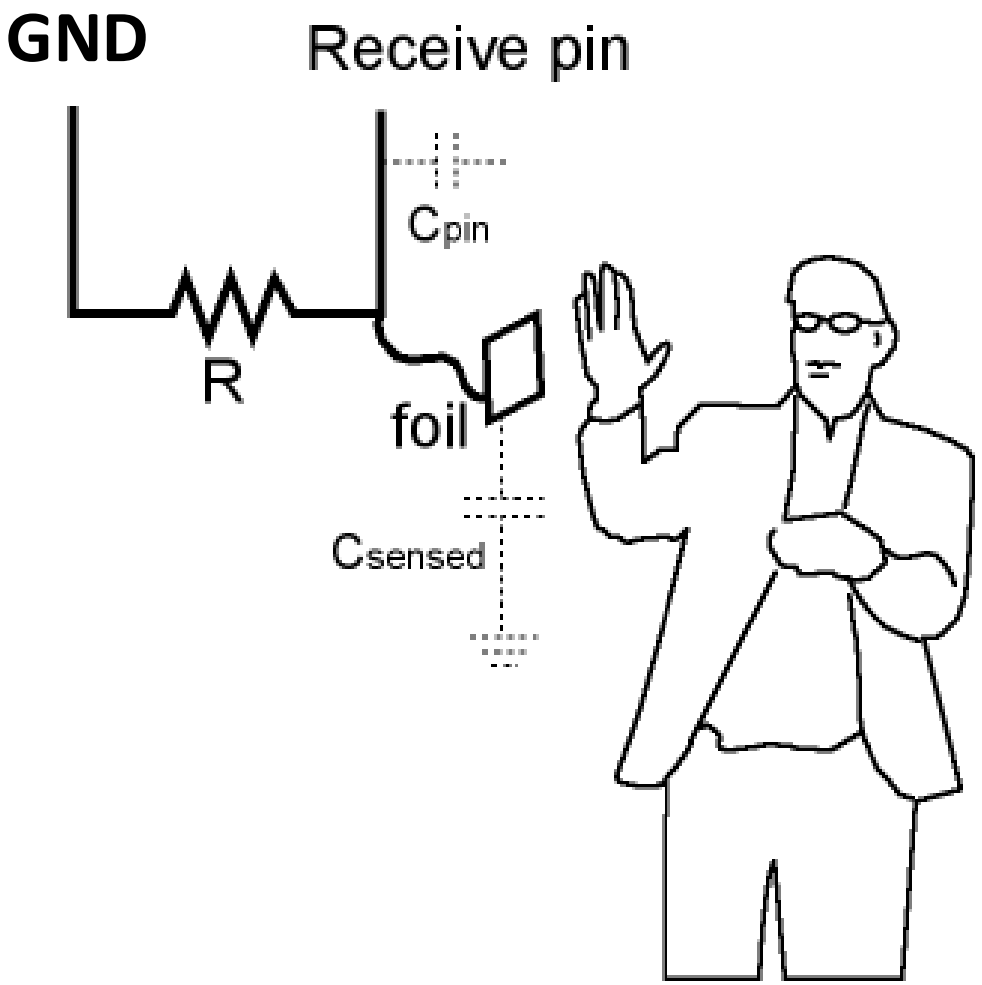
Hardware

- Breadboard
- Adafruit Circuit Playground Express
- 5x PIC16 (smaller versions of the PIC microchip used by the dev boards)
- 36x Homemade Capacitive Touch Sensors
  - Each needs 1 tin foil tab,
  - And one piece of wire
- Cardboard base (is carboard hardware?), again homemade, which means...
- Hot glue

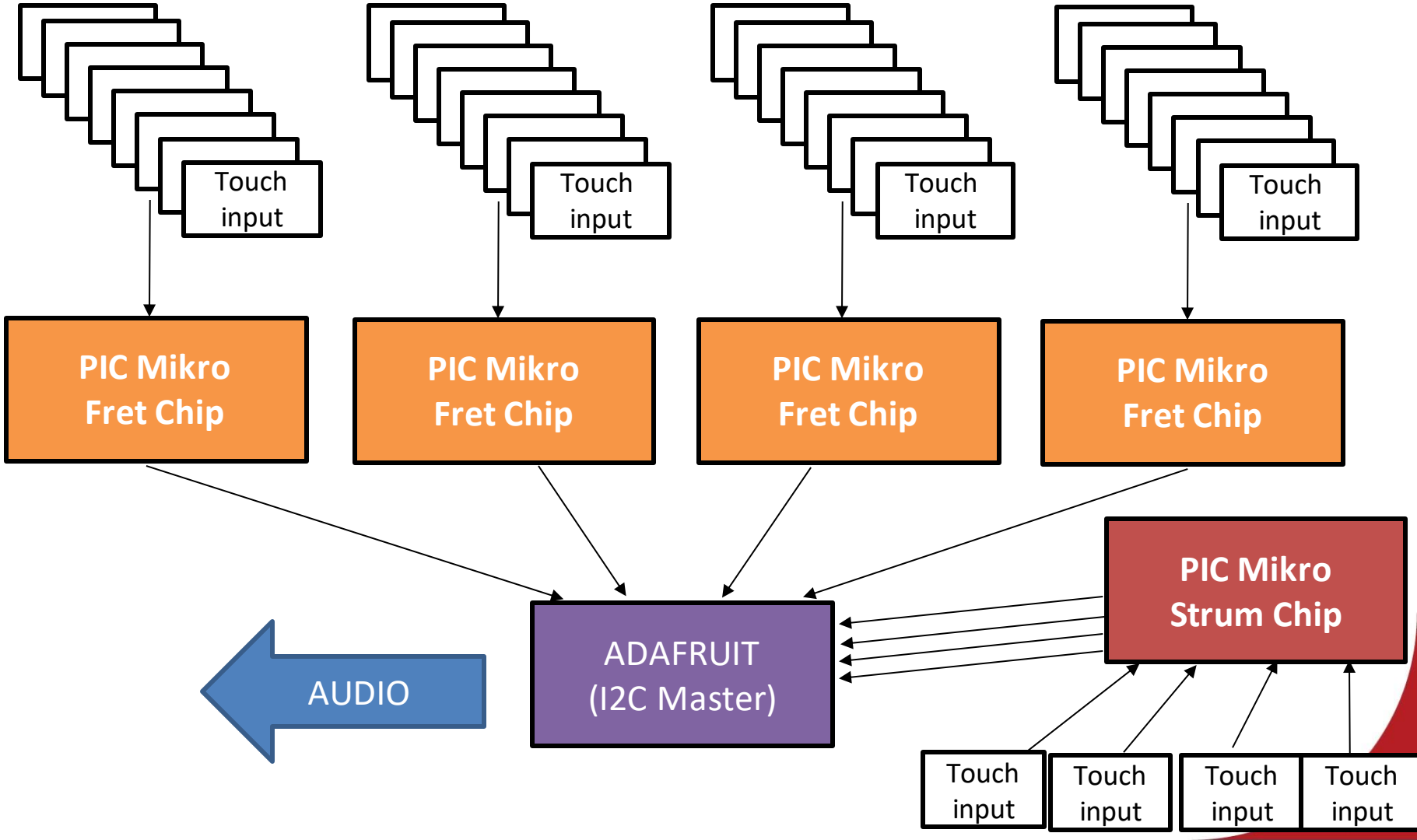
Software

- MikroC for PIC
- CircuitPython for Adafruit

DESIGN: Homemade Capacitive Touch Circuit



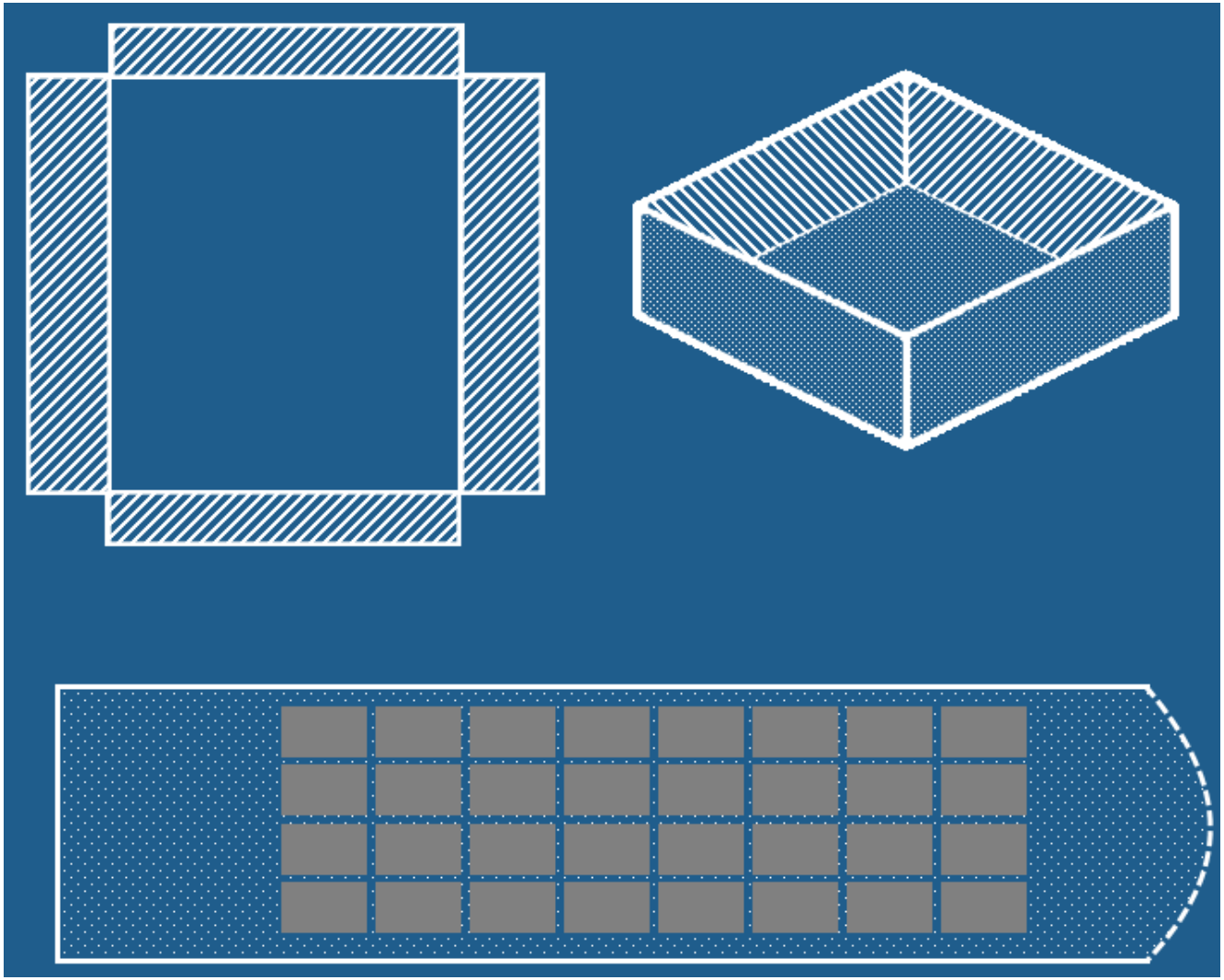
DESIGN: System Interconnections (Module layout)



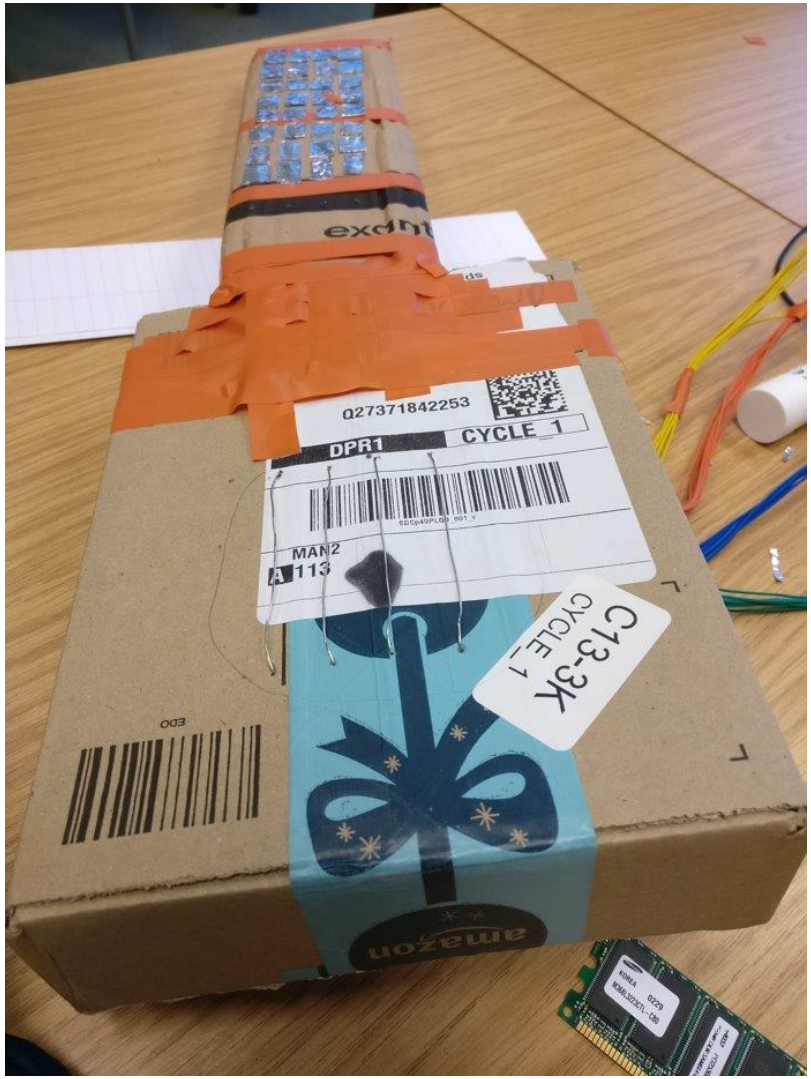
DESIGN: Guitar Body

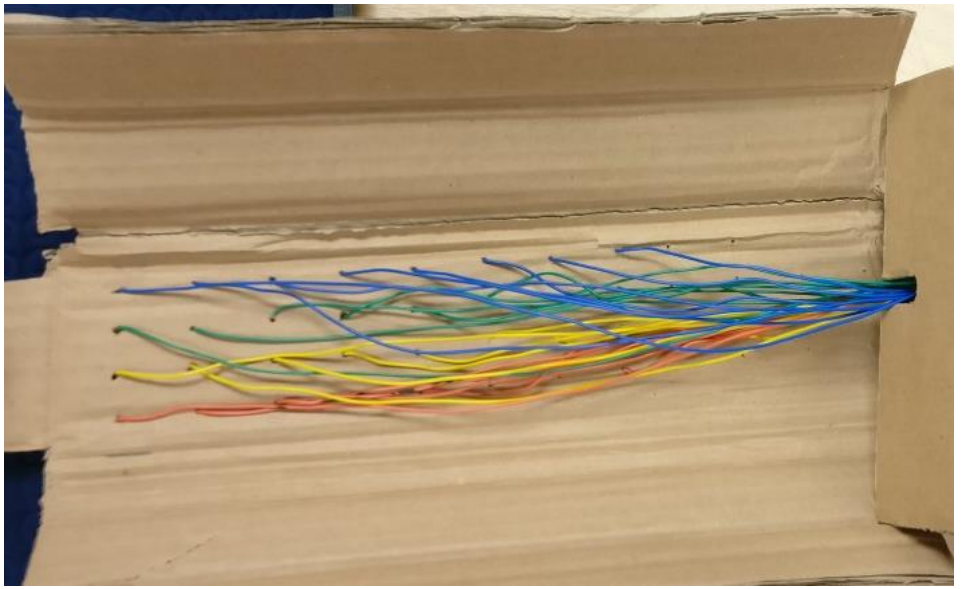
DESIGN: Guitar Body

DESIGN: Guitar Body



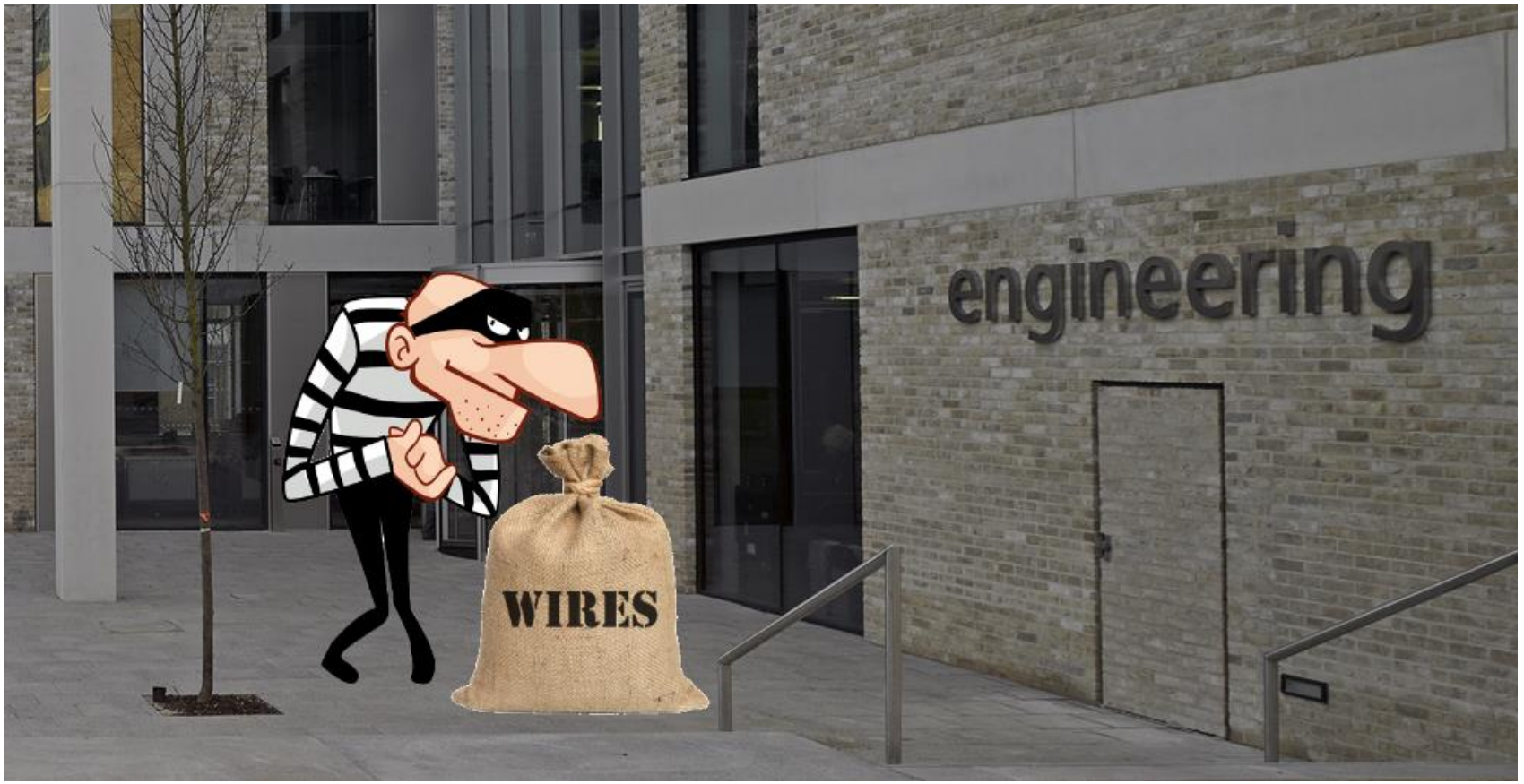
DESIGN: Guitar Body





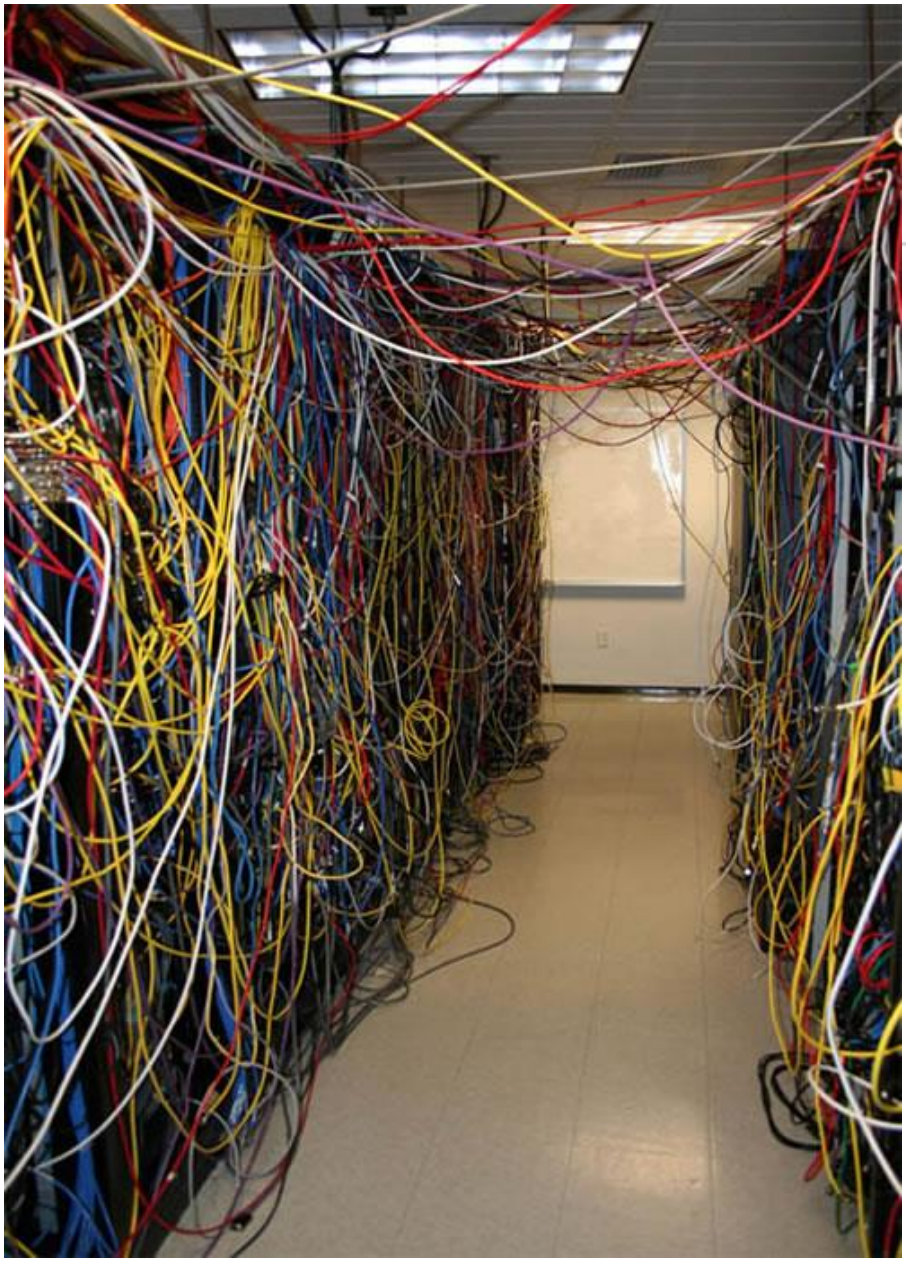
DESIGN: Guitar Body

Ethical Sourcing of Parts.



Lancaster University

Lancaster University



Lancaster University

Challenges

- **Creating** reliable capacitive touch inputs with only wiring, fuses, and tin foil plates
- **Cohesively making use of** different languages (C, Python) and hardware such as breadboards, PIC Mikro chips and the Adafruit Circuit Playground Express (CPX)
- **Manually wiring** physical I2C communications between arbitrary hardware devices
- **Successfully "sourcing"** 18 meters worth of differently coloured wire from the engineering department (thank you engineering!)

Problems

Lancaster University

- **I2C communications –** High level API communicating with low level API
- **Capacitive Touch Sensitivity –** Pull down resistors on sensors required many levels of calibration.
- **Debugging –** debugging was near impossible as we were only able to check single bit values via lighting up an LED attached to the breadboard.
- **Wiring** - approximately 50 wires inside a small cardboard case can cause many issues (shorting, loose connections, complexity...) Processors in neck or body of guitar, weight distribution.

Lessons

Lancaster University

- **Think small and get bigger, rather than thinking big and getting smaller** – project started as a "Rube Goldberg" of different pieces of hardware, and we quickly scaled down to the guitar project in week 2
- **There is no copy and paste in real life**. If your project needs 32 home made capacitive touch inputs, you will need to *build, test and integrate* a capacitive touch sensor 32 times.
- **Build for modularity.** If you set up a test circuit for one touch sensor without designing it to work for 8 later, you will need to redesign the entire circuit.
- **Managing many concurrent pieces of coursework is hard.**
- **Select your level of abstraction carefully**. The nature of the hardware we were working with meant that we were working with one very abstract API (CircuitPython) and one very near the metal one (MikroC). Both of these paradigms caused problems:

Demo and Questions

Lancaster University