

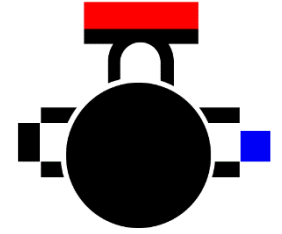
Hull Pixelbot and the IoT

All about Azure Internet of Things, Chatbots and the Hull Pixelbot

Rob Miles

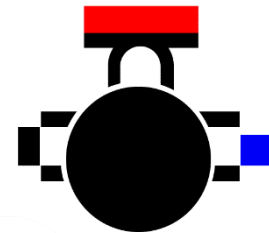
www.robmiles.com

Overview

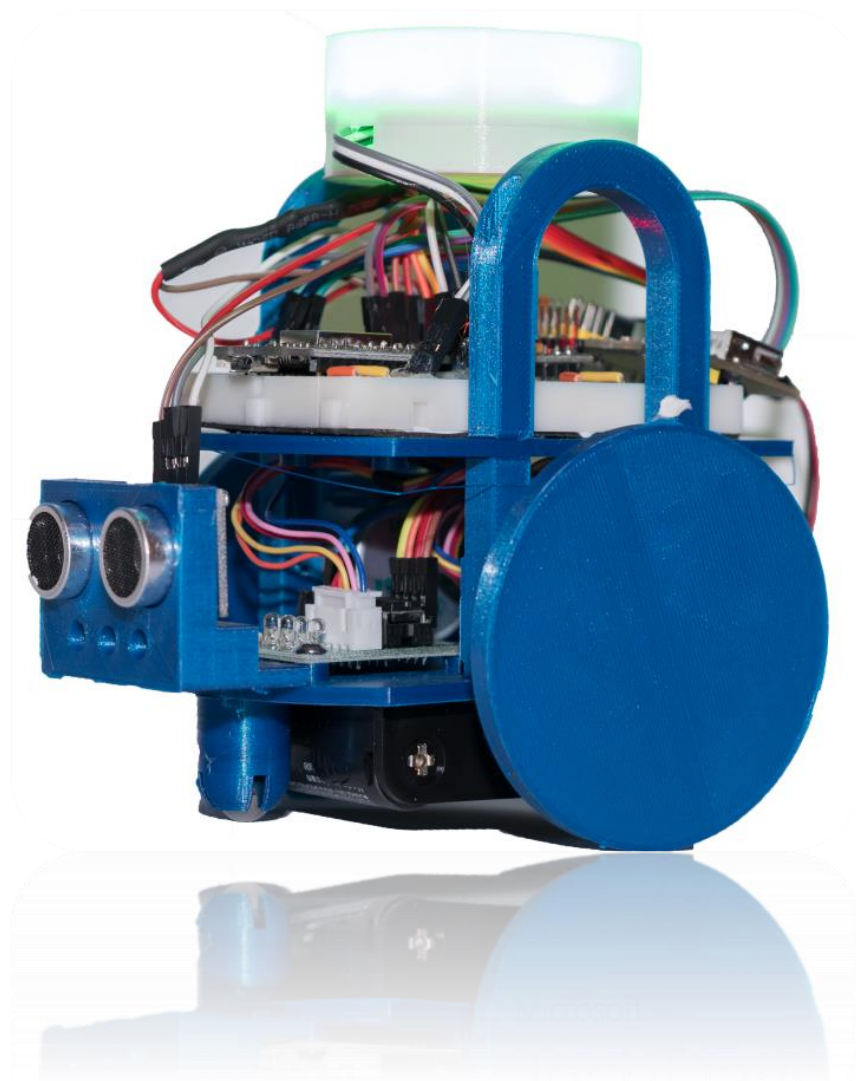


- Creating a robot – HullPixelbot
 - Robot Designs
 - HullPixelbot Code
- Connecting a robot
 - Web hosting
 - Azure IoT Hub and MQTT
 - Chatbots and robots
- HullPixelbot and the Future

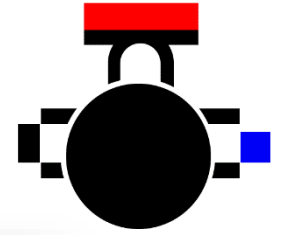




Building a robot

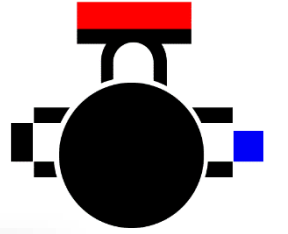


Why build a robot?



- It doesn't cost much
- You learn a lot
- It's fun

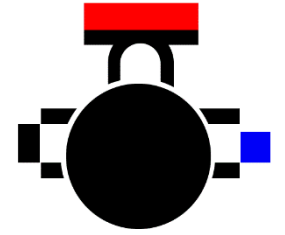
Why build a robot?



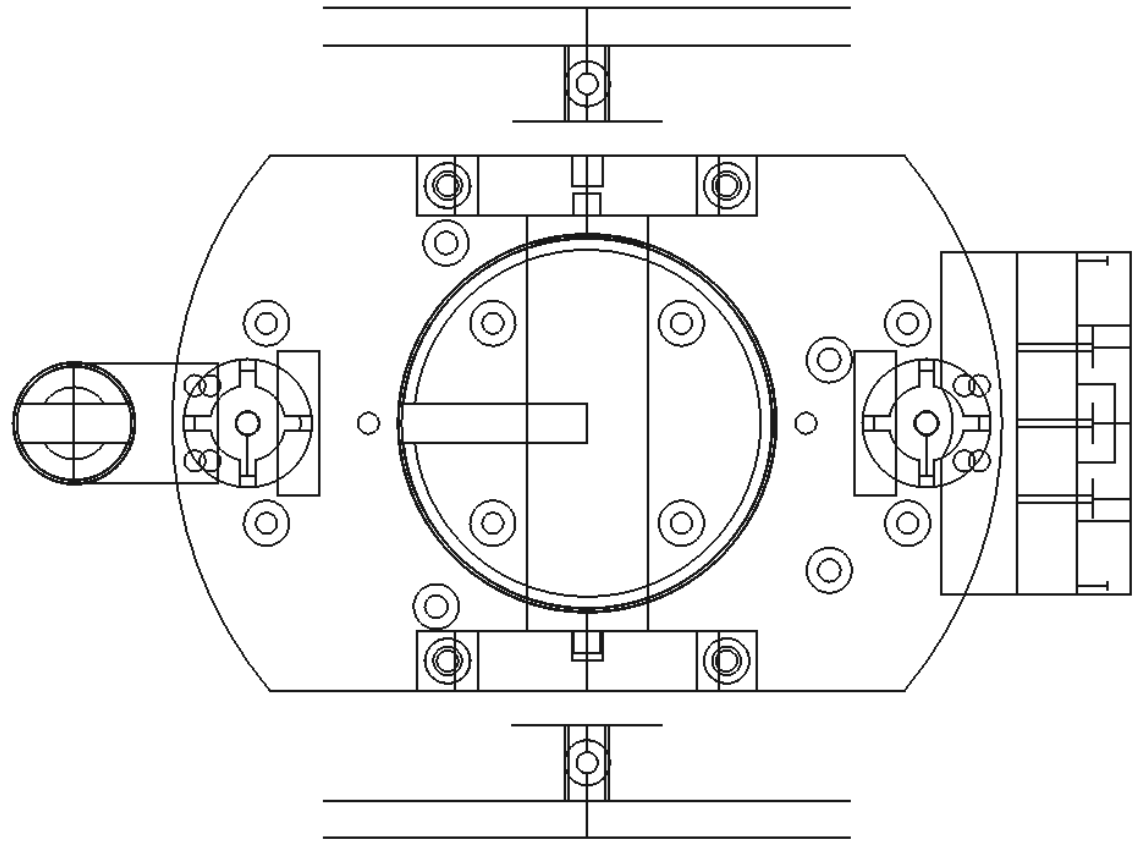
- It doesn't cost much
- You learn a lot
- It's fun

It gives you something to talk about at interviews

Design constraints

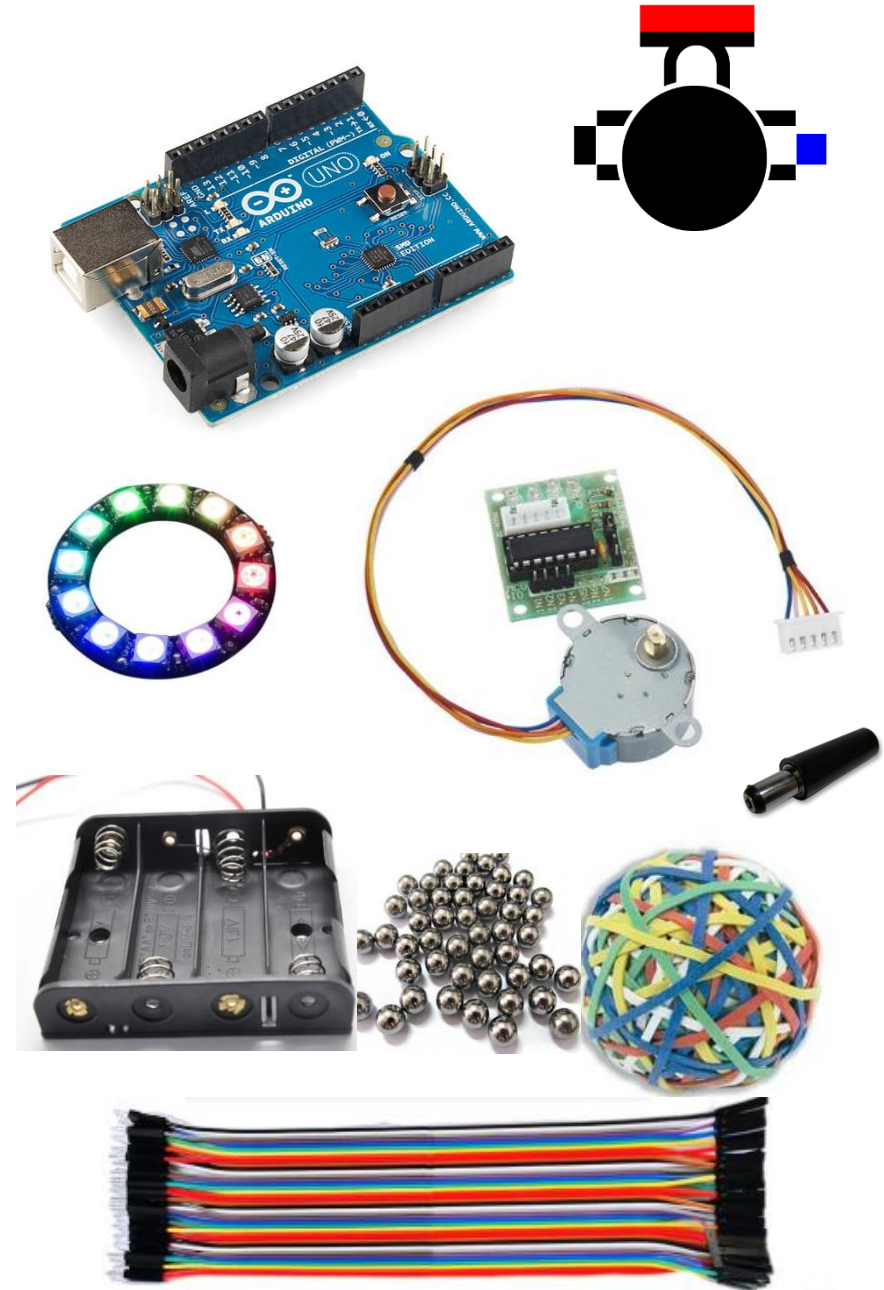


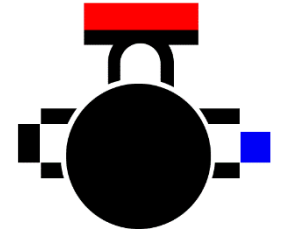
- Cheap to make
- Easy to build
- Extensible
- Connectable
- Cheap to make



Cheap to make

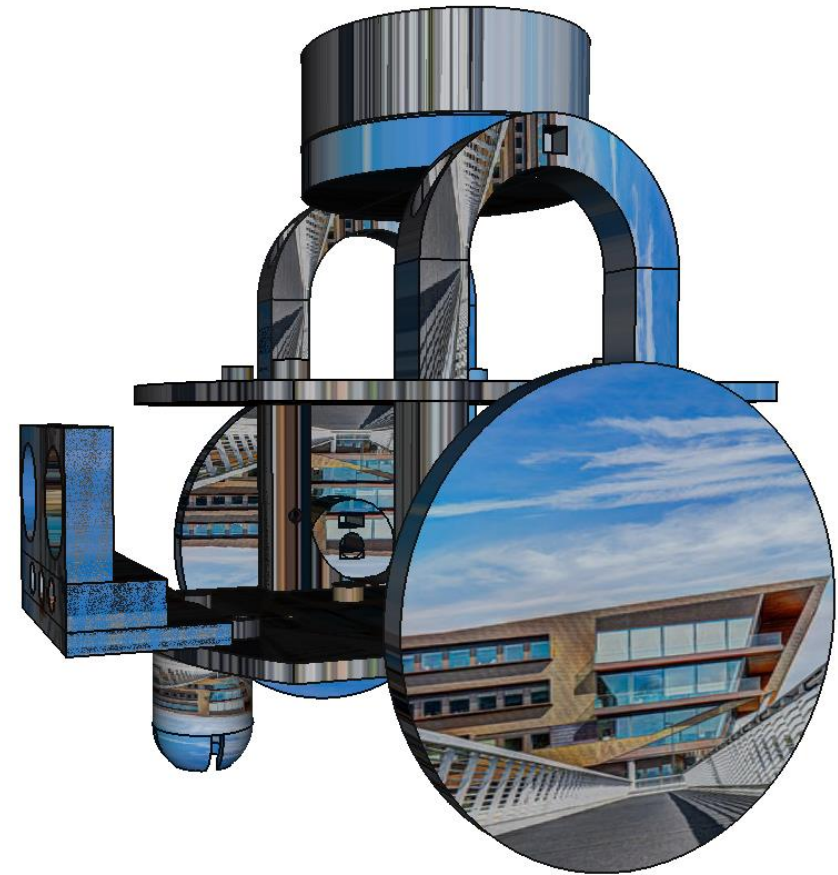
- Arduino processor
 - Less than five pounds
- Cheap stepper motor
 - Around one-fifty each
- Cheap pixel ring
 - Around one-fifty
- Distance sensor, battery holder, cables and nuts and bolts add around four pounds
- You can get the electronics for around ten pounds our so: www.aliexpress.com

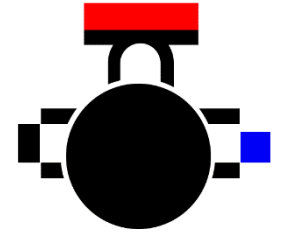




Easy to build

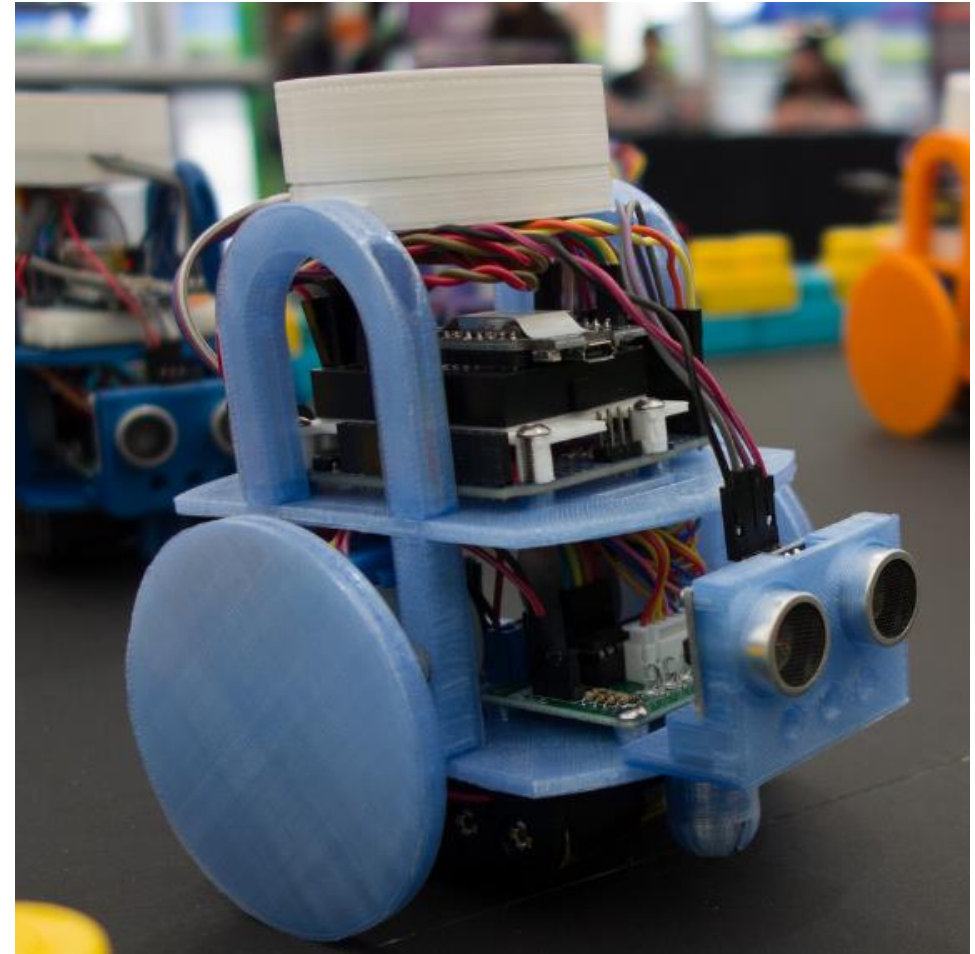
- All the parts are easy to create with a 3D printer
 - No nasty overhangs or tricky bits
- The robot just fits together using 3mm bolts
- You can design your own platform if you wish
- All the parts are flat, so they could be cut out of thin wood or Perspex



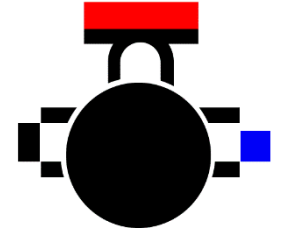


Extensible

- You can add any sensors that you like to the platform
 - Laser rangefinder
 - Temperature
 - Light sensor
 - LCD panel
 - Switches
- There are ready-made Arduino libraries for all these that you can pick up
- Or you can just build more robots...



Programming the HullPixelbot

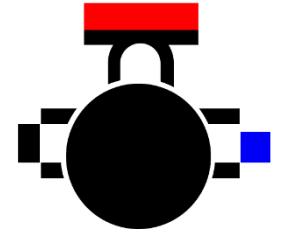


- You can write robot programs in C++ using the Arduino environment or Visual Studio using Visual Micro
- There is an Arduino library that you can use to drive the robot motors
- You can add your own behaviours based on the sensors that you add

A screenshot of the Arduino IDE interface. The window title is "FirstDrive | Arduino 1.6.11". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar shows icons for saving, running, and uploading. The main editor area displays a C++ program named "MotorDrive.h" with the following code:

```
6 #include "MotorDrive.h"
7
8
9 void setup()
10 {
11     setupMotor();
12 }
13
14
15 void loop() {
16     moveForwards(400);
17     turnLeft(180);
18     moveForwards(350);
19     turnRight(180);
20     moveForwards(400);
21
22     moveBackwards(400);
23     turnRight(180);
24     moveBackwards(350);
25     turnLeft(180);
26     moveBackwards(400);
27
28
29     stopRobot();
30 }
31
32
```

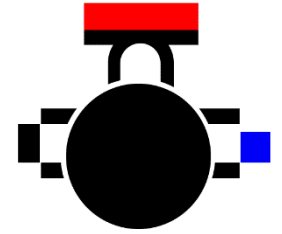
The status bar at the bottom indicates "Arduino Pro or Pro Mini, ATmega328 (3.3V, 8 MHz) on COM4".



HullPixelbot Code

- HullPixelbot Code was created to make it easy to control a robot with simple, text based, commands
 - It's a bit like Logo languages from a while back
- You can give the robot single commands and it will obey them
- The commands are interpreted by software running on the Arduino

```
#Just flash the lights
CLtop
PC255,255,0
CD10
PC0,255,255
CD10
CJtop
```

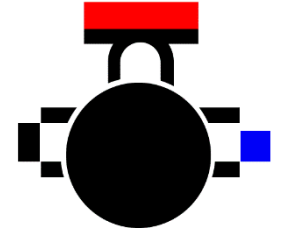


HullPixelbot Code

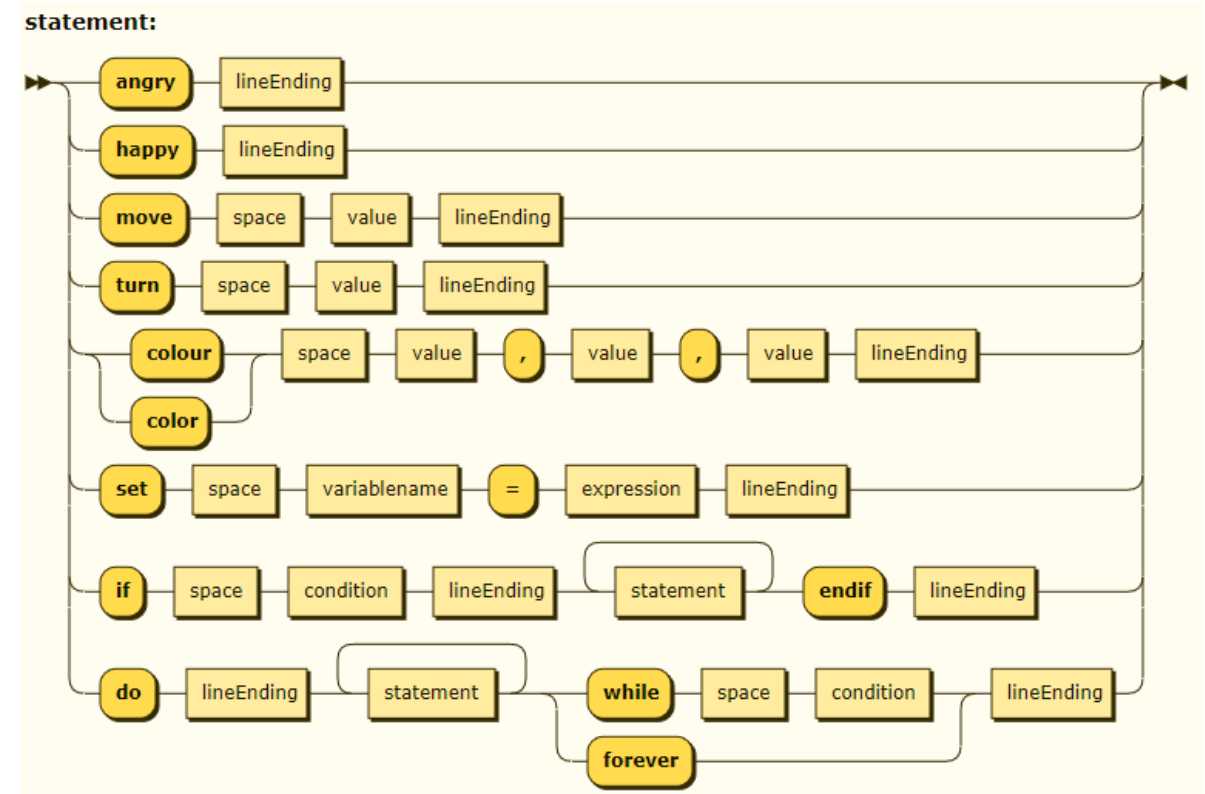
- Each command has a two letter identifier and is followed by a number of parameters
- This makes decoding the program very easy
- The code is retained inside the Arduino in its EEPROM
- When the Arduino starts the program runs

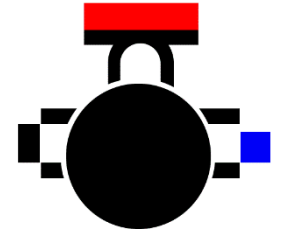
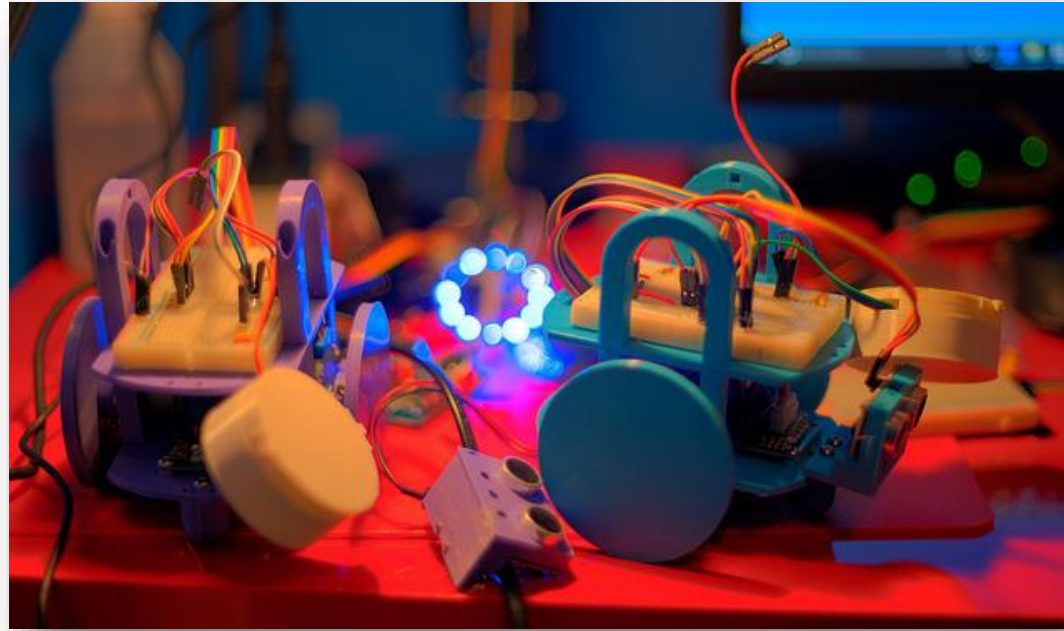
```
#Just flash the lights  <<comment
Cltop                    <<label
PC255,255,0             <<Pixel colour set
CD10                    <<delay
PC0,255,255             <<Pixel colour set
CD10                    <<delay
Cjtop                   <<branch
```

HullPixelbot Script



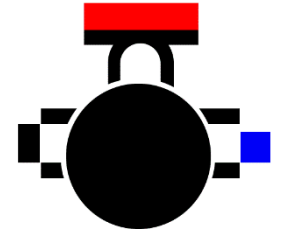
- I'm working on a more descriptive scripting language
- Statements will be translated into Hull Pixelbot code
- This will still happen inside the robot
- Got variables and conditional execution working



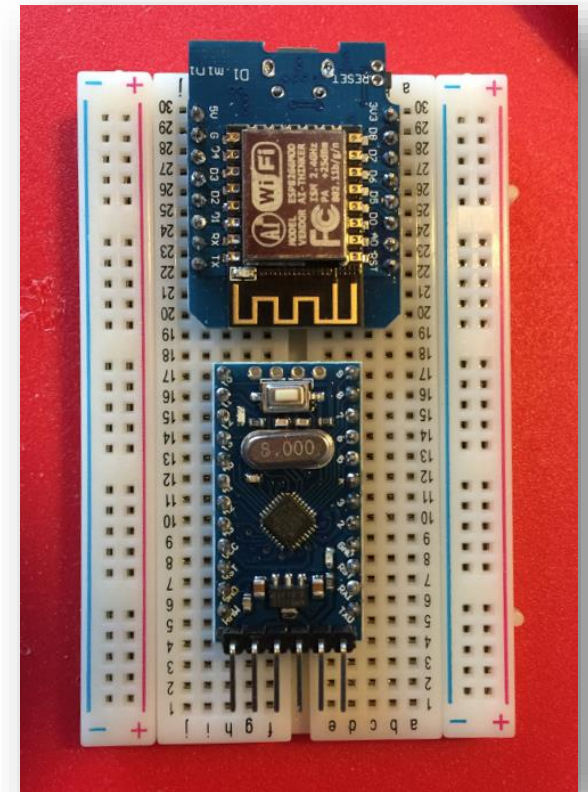


Connecting a robot

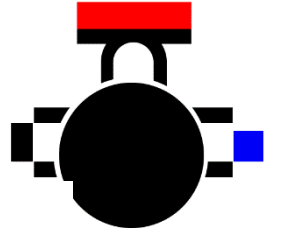
Dual processor robots



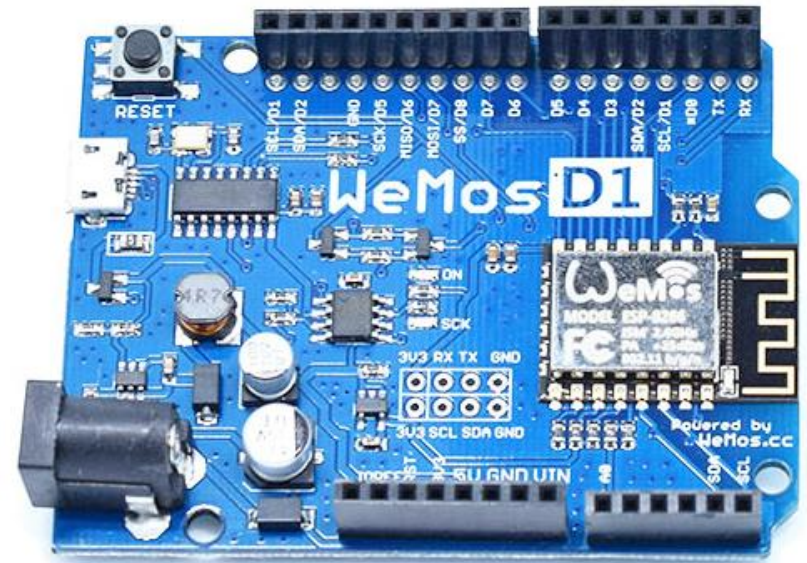
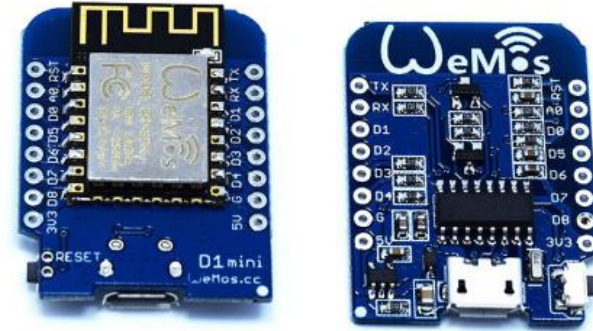
- Arduino Pro and ESP8266
 - One version of the HullPixelbot actually has two processors
 - Arduino for the input/output and motor control
 - Wemos D1 mini for the connectivity
 - This is a great way to create i/o heavy devices
 - Use a serial connection to pass commands between the two
 - For most simple systems you only really need a single device
 - But the Arduino Pro mini only costs around a pound....

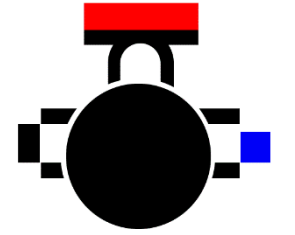


WiFi and the esp8266



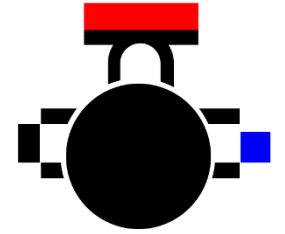
- Lots of WiFi options
 - WiFi client over a serial port
 - Fully programmable in C++
 - WiFi access point and web server
 - Support for UDP, TCP, secure sockets and mDNS
 - Very easy to use with many examples
- Making a connected client device
 - Lots of ways to do this
 - Web server, web sockets, MQTT, LAN, Access Point





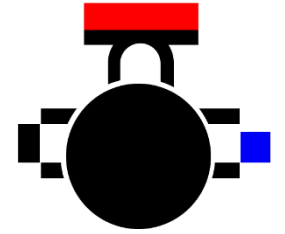
An esp8266 as a web server

- Create an embedded web server on a network:
 1. Select the Wemos D1 R2 and D2 mini platform
 2. Select the **ESP8266WebServer>HelloServer** example
 3. Set the SSID and the password in the code
 4. Deploy the program
 5. Connect via a terminal to view server output
 6. Connect via device on same subnet: esp8266.local using MDNS
- C++ methods are fired to deal with web requests
- This makes it possible to use web protocols to do just about anything with the device



An esp8266 as an access point

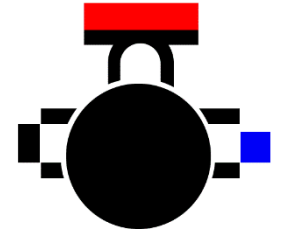
- Create an embedded web server on its own access point:
 1. Select the Wemos D1 R2 and D2 mini platform
 2. Select the **ESP8266WiFi>WiFiAccessPoint** example
 3. Set the SSID and the password in the code
 4. Deploy the program
 5. Connect via a terminal to view server output
 6. Connect a device to the access point. Browse: <http://192.168.4.1>
- This is a popular way to perform initial device configuration
 - Device hosts a web site that allows the entry of the WiFi parameters which are then stored in EEPROM for future use



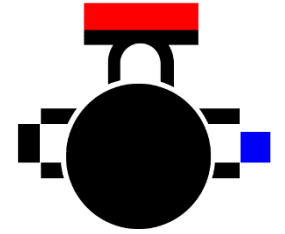
Esp8266 as an access point

- Only really supports one client at a time
- Can use websockets to communicate with the device
- The esp8266 also contains an internal file store that you can use to hold html files and other resources
- There is support for over the air (OTA) updates via WiFi
- Will support HTTPS connections too

Message Queue Telemetry Transport



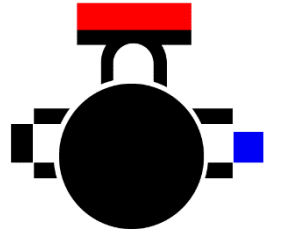
- MQTT is a way to connecting sensors to endpoints
 - It has a publish/subscribe architecture
- The communication can run over serial or WiFi and is based on a simple packet structure
- People have different opinions of how good it is, but it is very popular and also supported by the Microsoft Azure IOT Hub
- It also runs (surprise surprise) on the esp8266
- It is a great way to create cheap, connected, sensors



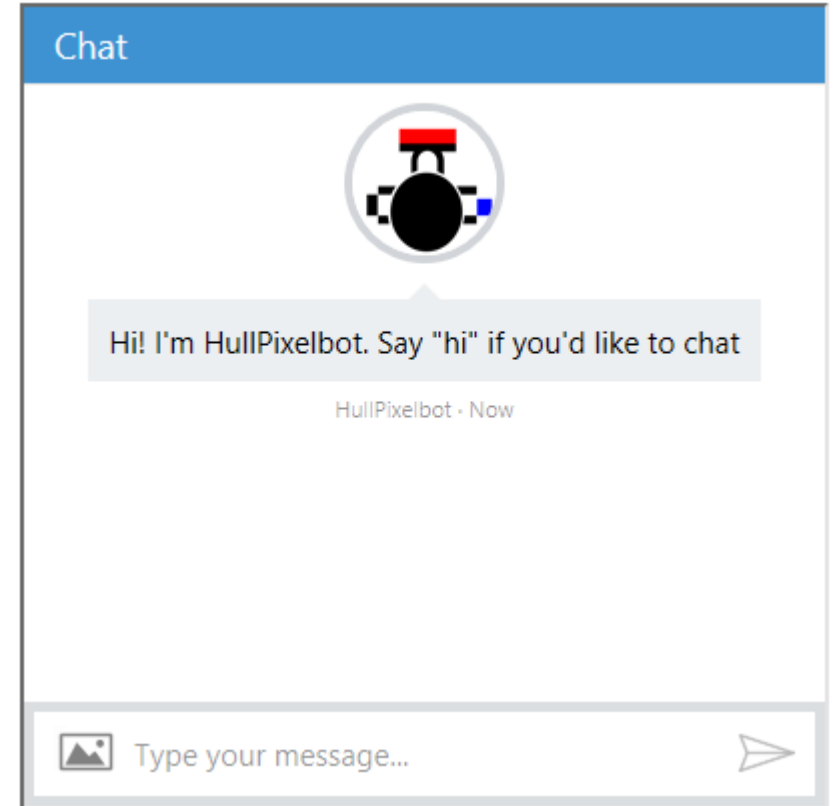
Azure, MQTT and the esp8266

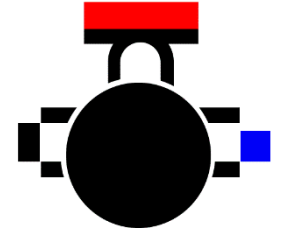
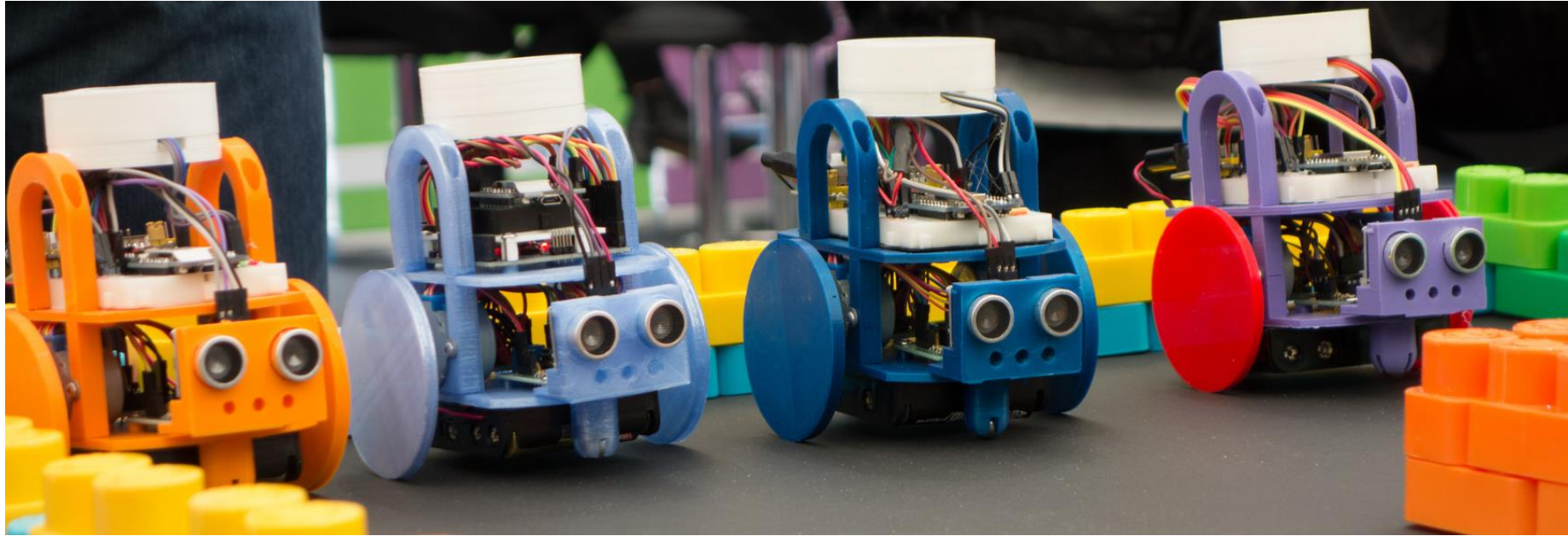
- MQTT PubSubClient for esp8266
 - I'm using the PubSubClient for esp8266 available at <https://github.com/knolleary/pubsubclient>
 - It needs to be modified for Azure:
 - Azure uses secure sockets, a different port and has larger packets
 - You can find out how to set everything up here:
 - <http://www.radupascal.com/2016/04/03/esp8266-arduino-iot-hub>
- Azure IoT Hub
 - The Azure IoT Hub will respond to MQTT messages
 - These can be passed on to your backend Azure applications and Azure applications can target MQTT devices

Chatbots and Robots

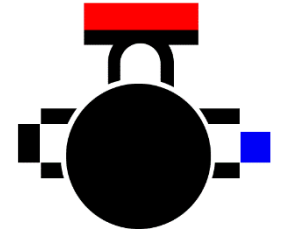


- I've built a Microsoft Chatbot framework that connects users to individual robots hanging off Azure IoT Hub
- It allows users to send commands to the robot and receive responses back
- I use it for robot racing and programming exercises





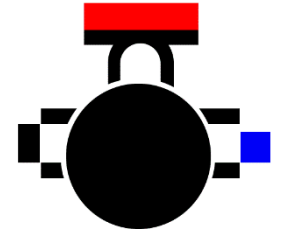
HullPixelbot and the Future



What next?

- I want to get a lot of HullPixelbots together and try to make some art
 - Ideally I want 100 or so at an summer event
 - I want it to be part of Hull City of Culture
- Owners should be able to bring their HullPixelbots along and take part in some activities
- We can film everything and take some long exposure pictures
- I'm working on tracking the robot positions using cameras
- We'll be working on these at the c4di hardware group

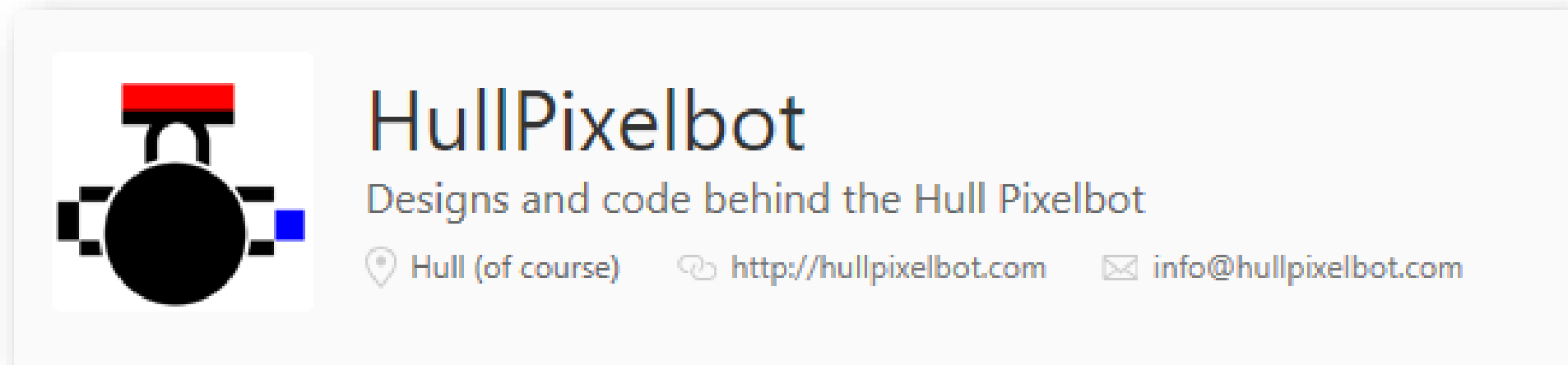
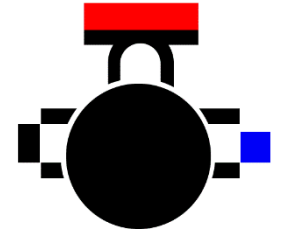
C4di hardware meetups



- We meet up every couple of weeks or so
- Everyone is welcome
- Sign up here:

<https://www.meetup.com/Hull-Digital-Hull-Open-Coffee/>

HullPixelbot on Github



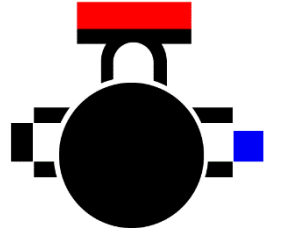
The image shows a GitHub profile card for HullPixelbot. It features the HullPixelbot logo on the left, which is a black circle with a red rectangle on top, a blue square on the right, and four black lines extending from the sides. To the right of the logo, the name "HullPixelbot" is displayed in a large, bold, blue font. Below the name, the description "Designs and code behind the Hull Pixelbot" is written in a smaller, blue font. At the bottom of the card, there are three icons: a location pin, a link, and an envelope. Next to the location pin is the text "Hull (of course)", next to the link is "http://hullpixelbot.com", and next to the envelope is "info@hullpixelbot.com".

HullPixelbot
Designs and code behind the Hull Pixelbot

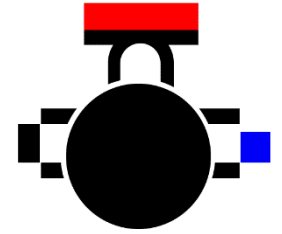
📍 Hull (of course) 🔗 <http://hullpixelbot.com> ✉ info@hullpixelbot.com

- All the designs and the software, along with construction details are on Github
- Keep track of the project at **hullpixelbot.com**

Win an esp8266 device



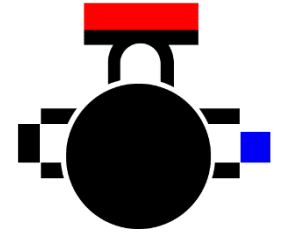
- If you want to get started I've got Wemos D1 R2 to give away
- This allows you to get started creating connected applications
- Just answer this simple question:



Win an esp8266 device

- If you want to get started I've got Wemos D1 R2 to give away
- This allows you to get started creating connected applications
- Just answer this simple question:

Wait for it.....

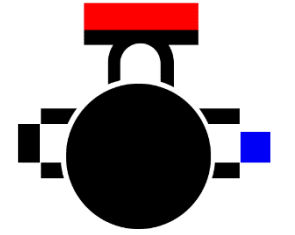


Win an esp8266 device

- If you want to get started I've got Wemos D1 R2 to give away
- This allows you to get started creating connected applications
- Just answer this simple question:

Wait for it.....

What does MQTT stand for?



Win an esp8266 device

- If you want to get started I've got Wemos D1 R2 to give away
- This allows you to get started creating connected applications
- Just answer this simple question:

Wait for it.....

What does MQTT stand for?

Message Queue Telemetry Transport

