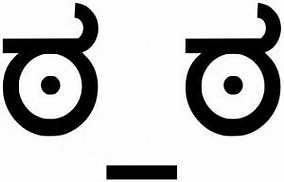
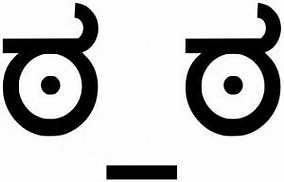
**Setting Up Patrick**



**MAKE ME MORE SERVICES!**

Project Patrick is a UWP application that can be run on a PC/Tablet/Phone or supported IoT device like a Raspberry Pi 3.

Clone the repository that can be found here on GitHub:

<https://github.com/PaulDFoster/Patrick>

Prerequisites are:

* Windows OS needs to be 10586 or higher
* The Speech Recogniser you use across devices will depend on what is installed on the device. We use Adaptive Coding to identify IoT Core from other Windows editions and change the speech recogniser accordingly on line 85. Edit En-GB to read En-US (Line 85) if you don’t have EN-GB on your PC.
* Use AllJoyn Explorer (Windows Store App) to test connection to the Alljoyn network connection. If you have an Alljoyn network, you need to have a supporting service running somewhere on your network. If you have a Raspberry Pi you can enabled the ZWaveAdapter headless service to do this for you.

The Billy solution uses a key word to identify actions it should interpret and execute. It runs a continuous dictation loop using the Windows 10 Speech Recogniser. On discovering the use of the CallSign the recognised text is sent to the LUIS language model to discover the intent. The query and any entities recognised by the LUIS model are returned with the discovered intent in the service’s JSON response.

You can (and should) define your own CallSign value on line 62. The call sign should be a name that is clear to pronounce, doesn’t rhyme easily, and you find gets recognised easily. Your mileage will vary if you have a strong accent or are using the generic en-US recogniser rather than a cultural specific one e.g. en-GB

Example phrases you can say:

What is the time?

Tell me a joke?

Look up <something, where something is a thing on Wikipedia. Note current speech recogniser build on the raspberry pi insider preview does not capitalise nouns correctly which upsets Wikipedia… try Look up machine learning

What is the weather doing today (will provide weather for Cambridge, UK)

If you have an Aeon Smart Switch on your Alljoyn network you can ask to turn it on.

Turn on the lights

Turn off the lights

Get the lights

Phases can be modified or added to in the LUIS language model utterances.

**Raspberry Pi Device Setup**

We use a Makibes 5” HDMI screen with our raspberry pi. This provides a nice self-contained device. You need to make the required display Config.txt changes on the OS SD card.

Config file for 5” HDMI screen:

#

# Video

#

framebuffer\_ignore\_alpha=1 # Ignore the alpha channel for Windows.

framebuffer\_swap=1 # Set the frame buffer to be Windows BGR compatible.

disable\_overscan=1 # Disable overscan

hdmi\_group=2

hdmi\_mode=1

hdmi\_mode=87

hdmi\_cvt 800 480 60 6 0 0 0

#max\_usb\_current=1

Wi-Fi setup – via device default IoT Core start up app. You’ll need a keyboard to enter the network key. Note after connecting to a network on the network list page, you don’t see any indication you have connected. Go back to the home page and see if you have an IP address.

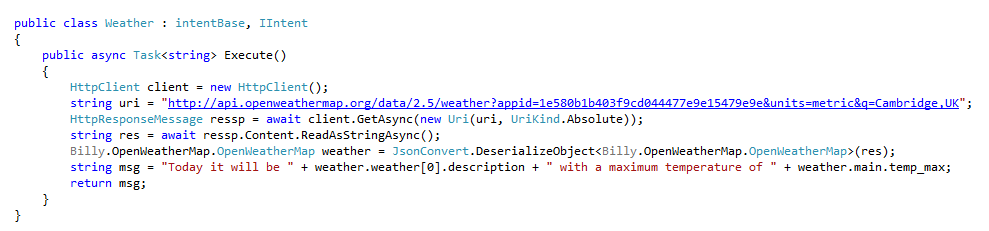
Device Portal – Use the device portal to set up your Raspberry Pi device from your PC. You can set up speaker and microphone volume on the Audio page. You can start up the ZWaveAdapter from the Apps page, and after the initial deployment you can set up the Billy solution to run at start up or when you want to. http:// (IP Address):8080

Note that Windows 10 IOT core requires a proper shutdown before removing power to stop corruption of the SD card.

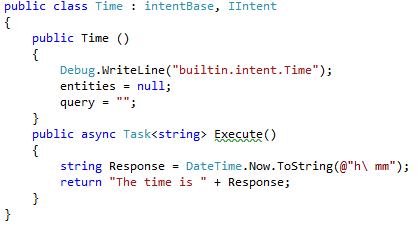
**Creating a New Intent in Billy**

It is really easy to add new service to the Billy solution in two steps:

1. Define the new Intent in the LUIS.AI model the Billy solution is linked too. See Editing LUIS.
2. Add a new Intent class with the same name as the Intent added in 1. The common name is the link by which the Billy app interprets what LUIS determined and the class to execute back on the Billy device.

**Example – Weather, Time**

Each intent is derived from intentBase class and IIntent interface. Only the execute method requires explicit implementation. It returns the message string that is spoken at the completion of the task.



**Rotate Intent**

The rotate intent drives a servo that is connected to signal pin 26 on the Raspberry Pi. The Billy Solution includes the PWMService which provides a PWM buss provider on the Raspberry Pi to drive servos. Servos should be wired up so that a common GND exists between the Servo battery and the Raspberry Pi. The servo signal cable is connected to Pin 26. The Servo positive cable is connected to the Servo battery.

Multiple servos can be controlled to create a JIBO like character by using additional GPIO pins on the raspberry pi.

**Editing LUIS**

<https://www.luis.ai/>

The default Billy solution uses a very simple LUIS application that is available for you to import into your own LUIS account service.

Alternatively, the LUIS service provides a Cortana Intent model which can be used. To use this model just reference the Cortana service but be aware that you need to implement the Intent class functions for the built-in Cortana intent model intents – and there are a lot of them. Alternatively, you can build your own simple model, as our example, and then grow the number of intent functions over time.

* In your LUIS Model, add an intent
* Enter sample utterances
* Train your model
* Publish your model and copy the URL into your Billy Solution