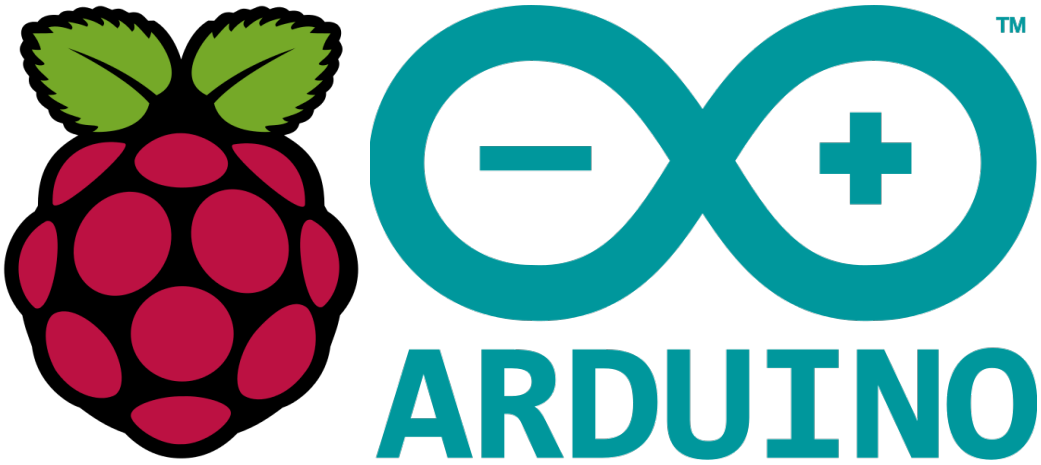
Brewduino

Andi Anderson Fraser George Adam McGhie
Aidan O'Grady Kristine Semjonova



CS413 Embedded Systems
Group One
Phase One Report
University of Strathclyde, Glasgow

We confirm and declare this report and the assignment work is entirely the product of our own efforts and we have not used or presented the work of others herein.

Signature: _____ Date: _____

Signature: _____ Date: _____

Signature: _____ Date: _____

Signature: _____ Date: _____

Signature: _____ Date: _____

Contents

1 Introduction

The initial scope of the assignment is to design and implement an embedded systems 'gadget' with the use of an Arduino and/or a Raspberry Pi.

After much deliberation and debate, we have decided to implement the Hyper Text Coffee Pot Control Protocol (HTCPCP)[3], with a system that allows the user to send requests for coffee over internet for either immediate or future consumption. A server will be implemented to handle the requests to be sent to our Arduino controlled coffee pot, which will then create coffee on demand. In addition, we hope to include the functionality of adapting the Barduino, a project from a previous year, to supply milk or other substances to a user's drink as well.

2 Assessment of Capabilities

2.1 Arduino

The Arduino is an open source microcontroller primarily aimed at electronics prototyping. It can read and interact with various components such as LEDs, motors, etc.

As it is a microcontroller it is designed to interact with analogue inputs of the outside world, which the Raspberry Pi does not have the ability to do as it can only deal with digital I/O (via GPIO). This allows easy collection of data through various sensors.

The Arduino Uno is the microcontroller we have decided to use for this project, as it meets the requirements for our project, listed below:

- Has sufficient size of onboard flash to hold the main program.
- Can be easily extended through the use of shields for additional hardware interface requirements.
- Uses 5V supply, meaning it does not require any power transformers to be compatible with other components (i.e. Raspberry Pi).

The primary disadvantages of the Arduino when compared to the Raspberry Pi is it's lack of network capabilities, and the Arduino's lack of processor power. The Pi allows us to run the necessary independent web server, and make it accessible to an external client. Due to this the Pi shall be do any computationally intensive tasks and use the Arduino to interact with the other hardware.

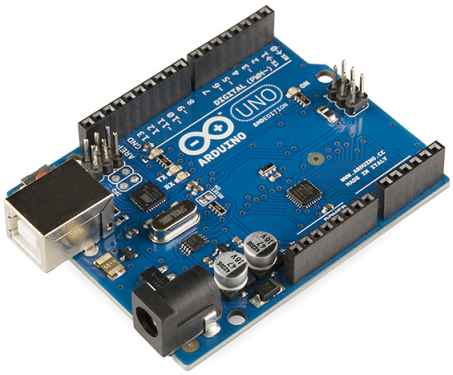


Figure 1: Arduino Uno.[2]

2.1.1 Specification

This based on the official specification sheet found here[1]:

Microcontroller	ATmega 318P
Operational Voltage	5V
Input Voltage	7-12V
Input Voltage (Limits)	6-20V
Digital I/O	14 (6 PWM)
Analogue Input	6
DC Current per I/O Pin	40mA
DC Current per 3V Pin	50mA
Flash Memory	32KB (0.5KB reserved for bootloader)
SRAM	2KB
EEPROM	1KB
Clockspeed	16 MHz

2.2 Raspberry Pi

The Raspberry Pi is a single-board computer with a small form factor (roughly the size of a credit card), which was created as an educational tool by the Raspberry Pi Foundation.

It commonly uses the Linux operating system, and has a Broadcom system on a chip, which includes an ARM1176JZFS, with floating point, running at 700 MHz, and a Videocore 4 GPU.It is also capable of 24 GFLOPS of general-purpose compute and features several texture filters and DMA infrastructure. The Pi's OS must be booted from a microSD card though an external storage device can be used afterwards.

The Pi is a more powerful device than the Arduino, since it is designed as a computer out of the box, and therefore and has more capabilities rather than just being a microcontroller. This makes it ideal for acting a middleman between the user and the Arduino.

The Pi will be used to run the Java Spring application allowing it receive requests from external clients. It can then handle their requests and interact with the arduinos to operate the Coffee machine and Brewduino. This means our most likely course of action will be accessing the Pi through its IP address in the user's browser.

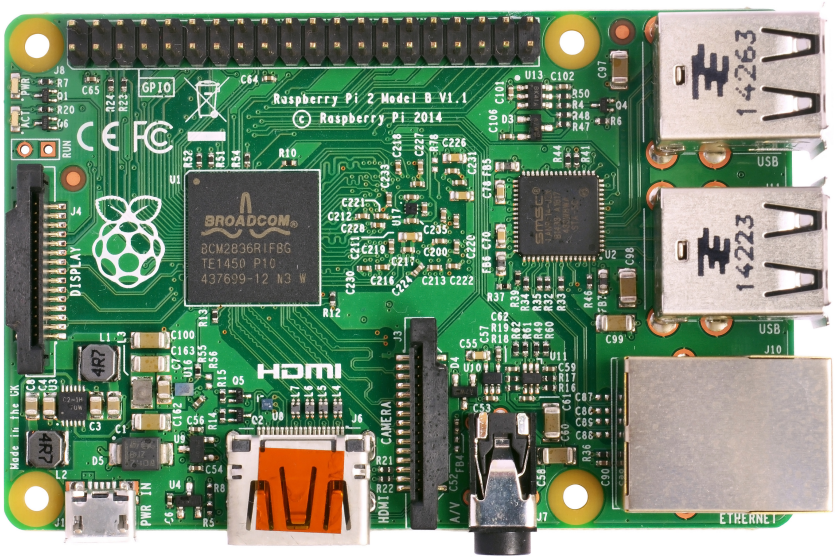


Figure 2: A Raspberry Pi Model B.[4]

2.2.1 Specification

System on Chip	Broadcom BCM2836
CPU	900 MHz quad-core ARM Cortex-A7
Memory	1 GB
Ethernet (Limits)	10/100 Mbits/s
USB 2.0	4 Ports
On Board Storage	MicroSD slot

3 Development Environment

The development environment that we shall use for the Arduino is the Arduino Integrated Development Environment (or Arduino Software (IDE)). For development on the Pi we will be using Raspbian with a Java Spring application acting as our web service for the system.

We all also be using a Git version control system in addition for easier transferring of work between group members. For the purposes of sharing other documents and files (for example, this report), a shared Google Drive folder was created to allow us all to add to each other's notes with ease before being formatted and organized properly.

4 Additional Resources

4.1 Coffee Pot

Since the HTCPCP is all about sending requests for coffee, a coffee pot is naturally required for the whole thing to work. We have decided on the Andrew James 1100 Watt DIgital Filter Coffee Maker.

Price: £27.99

Purchased From: Amazon UK

4.2 Barduino

As part of our project, we have found the potential of adapting the Barduino, a previous year's Embedded Systems project into our own, for the use of supplying the coffee with milk and other condiments if desired. Since the project owners have not claimed it, it has been deemed available for appropriation.

Price: £0.00

Purchased From: N/A

Figure 3: The Barduino

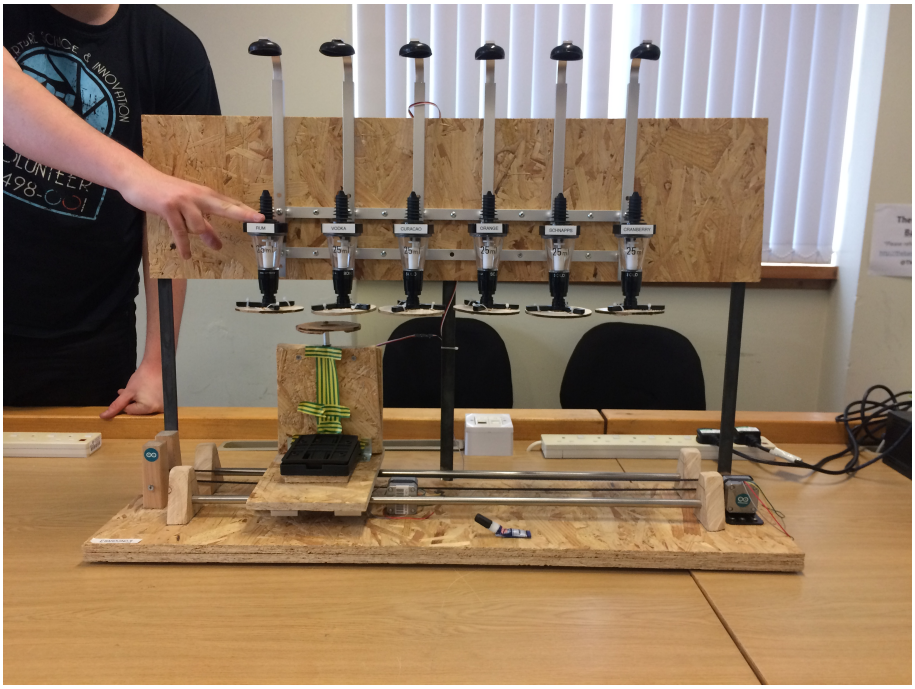


Figure 4: The Coffee Maker

5 Design

5.1 Overall

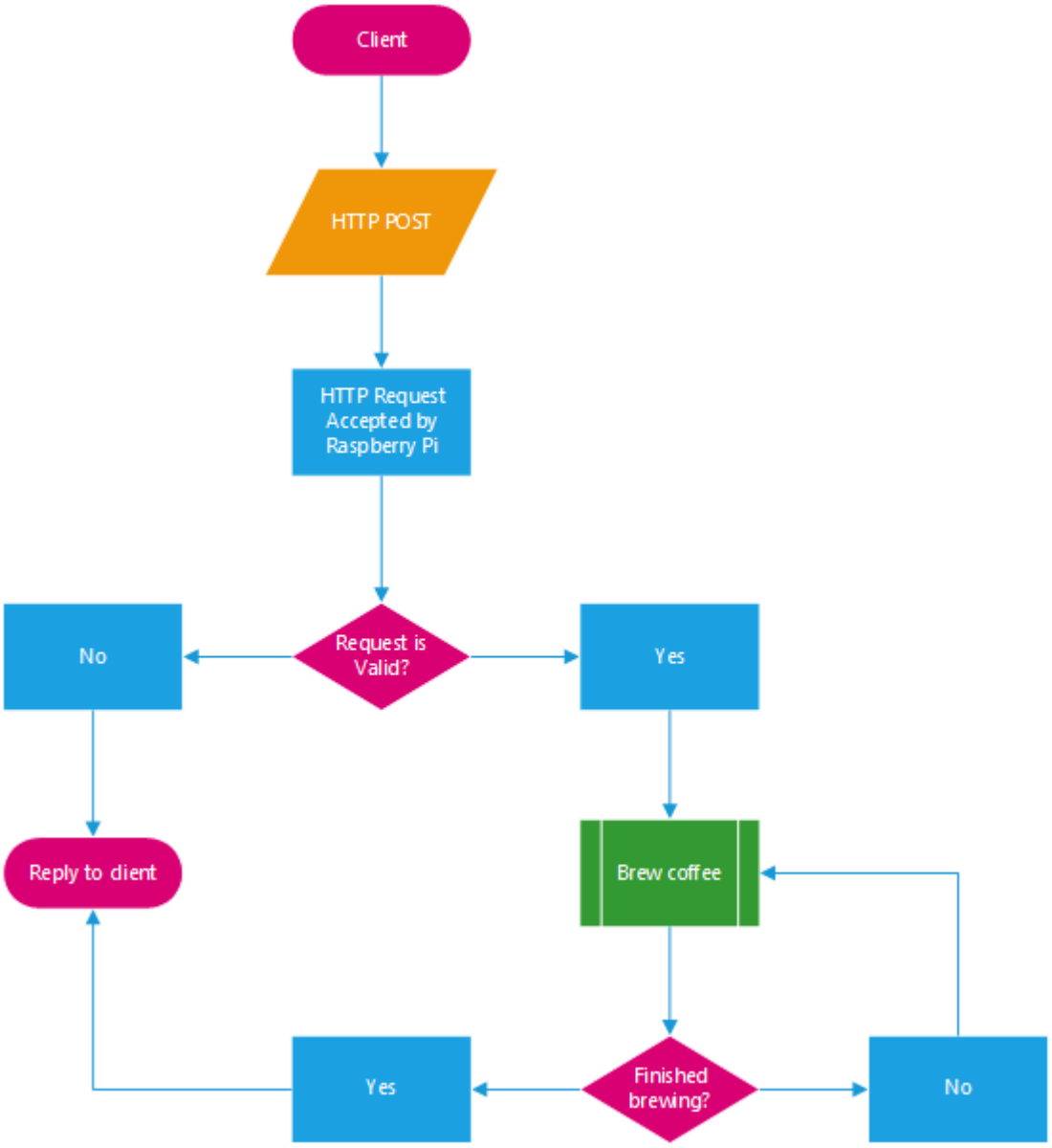


Figure 5: A flowchart defining how the overall system works

5.2 Hardware

5.3 Software

There are three primary components of the software side of this project. Those are:

- A website through which users can order coffee.
- A Java Spring application to handle the above requests and pass to Arduino.
- The Arduino software triggering the coffee brewing and serving.

The typical user will only be exposed to the website aspect of this stack, which will be able to provide all neccessary information through the website, such as when there is no coffee available.

5.3.1 Website

The website is relatively simple. In order to obtain coffee, all they need to do is complete the form that is provided to them. Given the current set up, the URL is likely to be off the form IP:8080/brewduino, where IP is the address of the Pi. The system will ideally provide the user information such as the amount of coffee available right now, and how many requests are currently in a queue to be filled.

5.3.2 Java Spring Service

The Raspberry Pi will be hosting a Java application utilising the Spring framework to allow for us to take the form submission and act upon it in Java. This has the advantages of allowing communication between Java application and the user, as well as between the application and Arduino.

The Java application is made up of four core classes, this system is a spin-off of Spring's guide on Handling Form Submissions[7] (due to the structure of this, a class diagram was deemed unsuitable for provision):

- Application
  - Main method runs the system here, through the Spring framework, making it an extermely simple class.
- Brewduino
  - This class acts a data structure representing the form the user fills out on the website. Each field in the class represents a part of the website's form.

- BrewduinoController
  - This class handles the requests sent from the user to the Pi, and acts upon them. It will validate that the input is OK, and then communicate with the Arduino to fulfill the request.
- Arduino
  - This class handles all communication with the Arduino, acting as a middleman between the Arduino and the BrewduinoController.
  - The Arduino class will make use of a suitable framework, such as JArduino[6] or ZXTX[5] to handle communication with Arduino.

6 Current Progress

When we started this assignment, the Brewduino was not our original concept. The initial idea was to create a Dalek (or Dalek like machine) whose camera could be viewed in a browser. However, we decided to go with the Brewduino project after it was suggested a couple of weeks later, since we felt it would be a more interesting project to tackle. This has had the

So far, we have acquired the coffee pot and commandeered the Barduino for our project. In addition, an early start has been made on the Spring application, since some development was done as part of the research into its feasibility and suitability for the Brewduino.

An attempt has been made on the disassembly of the coffee pot we are using to be connected to our Arduino, although that has not been completed (see the plan for the projected completion).

References

[1] Arduino. Arduino - arduinoboarduno. https://www.arduino.cc/en/Main/ArduinoBoardUno.

[2] SparkFun Electronics. Arduino uno r3. https://www.flickr.com/photos/41898857@N04/8406865680.

[3] L. Masinter. Rfc 2324 - hyper text coffee pot control protocol (htcpcp/1.0). https://tools.ietf.org/html/rfc2324, 1998.

[4] Multicherry. Raspberry pi: Model b v1.1. https://en.wikipedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new_%28bg_cut_out%29.jpg.

[5] Rxtx. Rxtx. http://rxtx.qbang.org/wiki/index.php/Main_Page.

[6] SINTEF-9012. Sintef-9012/jaruino. https://github.com/SINTEF-9012/JArduino.

[7] Spring. Getting started - handling form submissions. http://spring.io/guides/gs/handling-form-submission/.