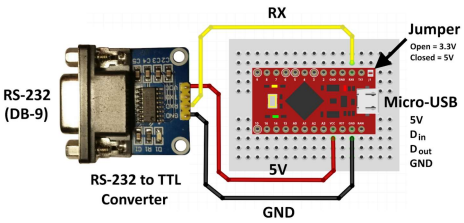
Unitec Institute of Technology

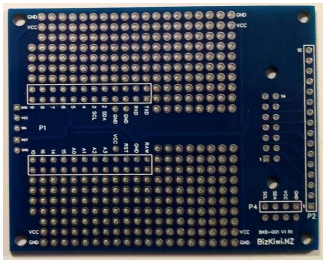
DE6102 – Engineering Project

Serial Port Monitor

"Making Fast Service Faster"

Project Report





Stephen Julian

Student ID: 1041664

Project Supervisor

Morgan Look

Acknowledgements

Special thanks for helping make this project possible go to:

- Morgan Look, my project supervisor and manufacturing tutor.
- Dr. Mirjana Bogosanovic, my project tutor.
- Dr. Nigel Yee, my microcontrollers tutor.

Thanks also go to Kham Hong Kok and Anthony Lai for their helpful advice and support.

Last but not least, thanks to Mark and Sarah and all of the staff at McDonald's Queenstown.

Abstract

This report is about the development of a device that enables the customers of a Quick Service Restaurant to be served their meals on the order of ~2 to ~6 seconds faster than previously.

For the fast food industry this can be considered a fairly significant margin of improvement.

There are four parts to this report:

1. Introduction
2. Project Design
3. Project Engineering
4. Conclusion

Part 1 introduces the project overview and the motivation behind the efforts.

Part 2 covers the existing order serving method and system at the restaurant as well as the improved order serving method and system.

Part 3 details the engineering of a series of prototype devices, each one building upon the outcomes and lessons learned from the previous one.

Part 4 summarises the outcomes of the project and looks at recommendations for the future.

Table of Contents

1 Introduction

1.1 Terms of Reference

This report documents the work of student Stephen Julian with the DE6102 Project paper at Unitec Institute of Technology. This work has taken place under the tuition of Mirjana Bogosanovic and the supervision of tutor Morgan Look.

1.2 Procedure

Information and content for this report was gathered from photographs, diagrams and screenshots taken during the project, content sourced from Internet websites.

1.3 Project Overview

The objective of this project was to build a Serial Port Monitor device capable of monitoring and non-invasively intercepting data sent between a PC and a printer connected via RS-232 serial cable. The device was required to parse intercepted text data for keywords and trigger events by writing data to a USB port connecting the device to a PC as the USB host.

1.4 Motivation

Automated monitoring of the content of data sent to printers has a wide range of potential applications in Quick Service Restaurants and many other customer service and other industries.

1.5 Market Research

There are various hardware RS-232 serial Input / Output (IO) monitors currently available in the market. These typically capture data transmitted along a serial cable and forward all of this data to a PC or server for analysis with protocol analyzer software.

Broadly the market is split into 3 categories including devices with data forwarding via:

1. USB interface.
2. Ethernet interface.
3. Bluetooth interface.

Market research was unable to find existing hardware capable of on-board protocol analysis (parse for key word/s of interest using delimiters) as they instead perform this function with external servers or PC hardware and software.

2 Project Design

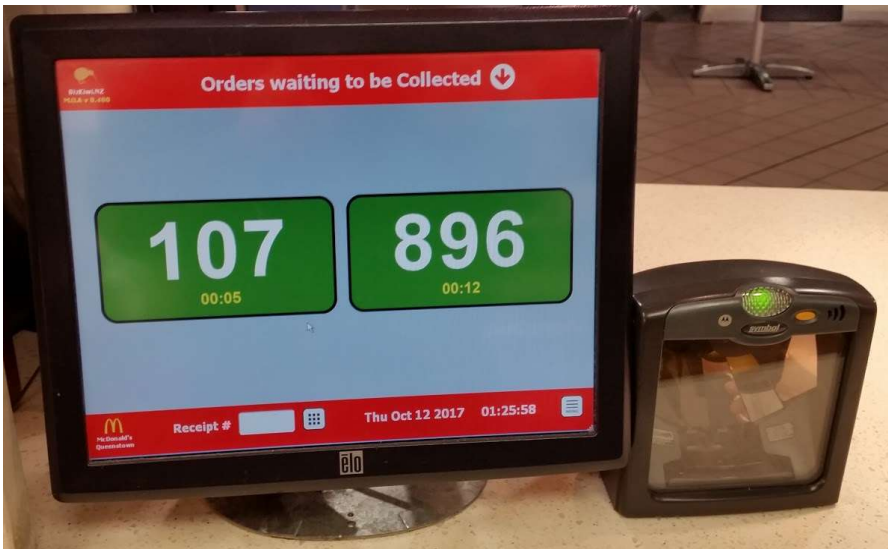
2.1 Original Method for Serving an Order (Figure 1)

Customer Order is Ready

Employee Presses SERVE Button

A Pick Slip Prints from the Printer

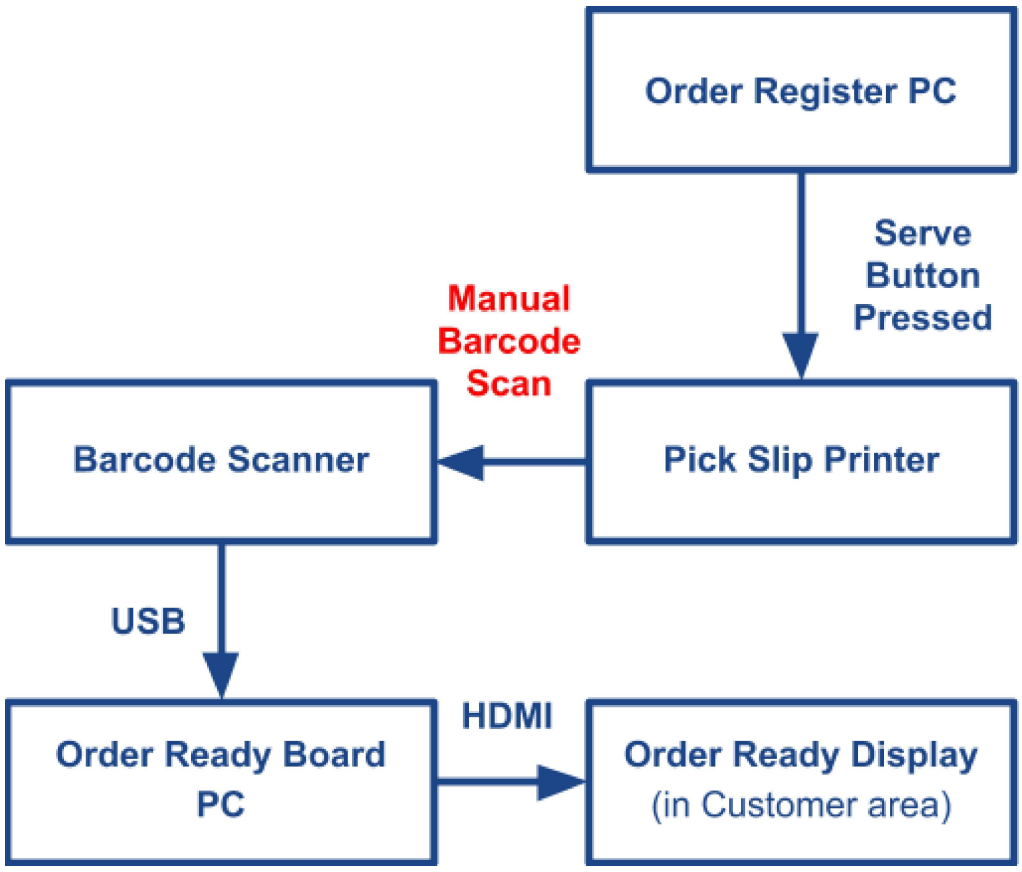
Employee needs to **MANUALLY SCAN** the Printed Barcode



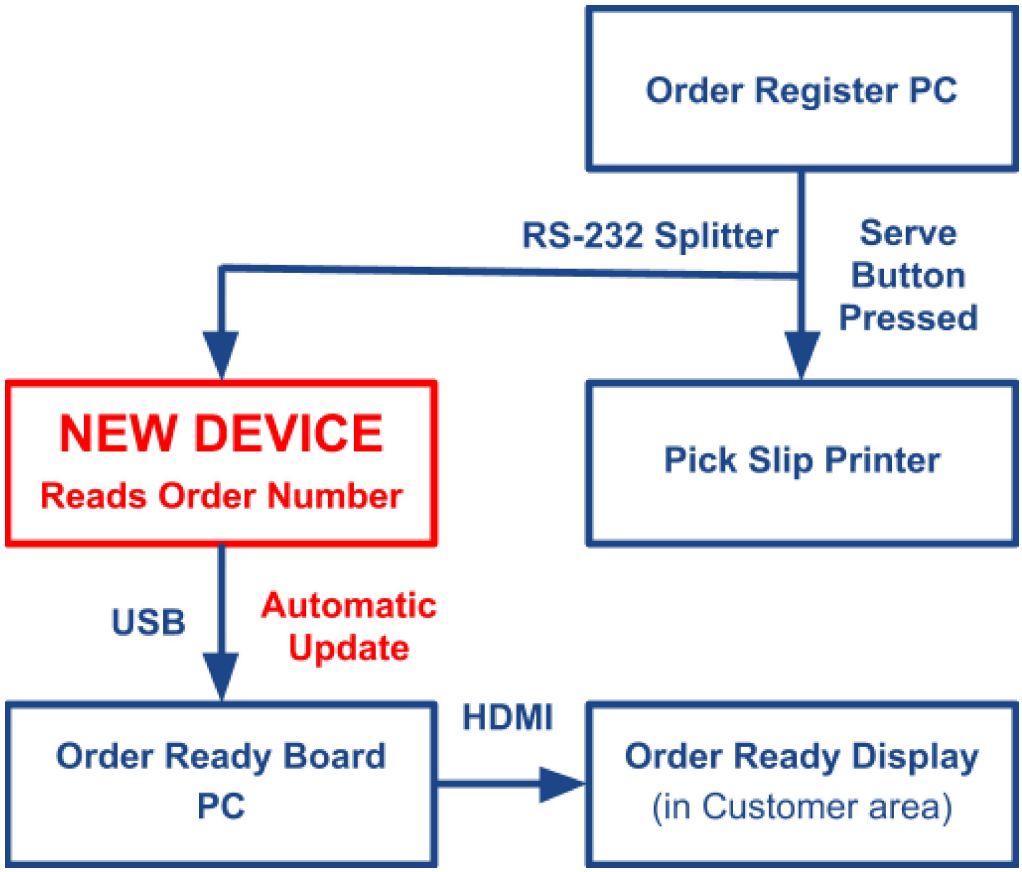
Order Number appears on Display in Customer Area.

Customer sees Order is Ready to Collect

2.2 Original System (Figure 2)



2.3 Improved System (Figure 3)



2.4 Improved Method for Serving an Order (Figure 4)

Customer Order is Ready

Employee Presses SERVE Button

A Pick Slip Prints from the Printer

THE NEW DEVICE
Reads Order Number from the Pick Slip Printer Data Stream then Emulates a Barcode Scanner

NO SCANNING REQUIRED!

Order Number Then Appears on Large Screen in Customer Area. Customer sees their Order is Ready to Collect

THE RESULT?
Customers Receive Orders ~ 2-6 Seconds FASTER!

3 Project Engineering

3.1 Prototype 1

3.1.1 Functions and Features

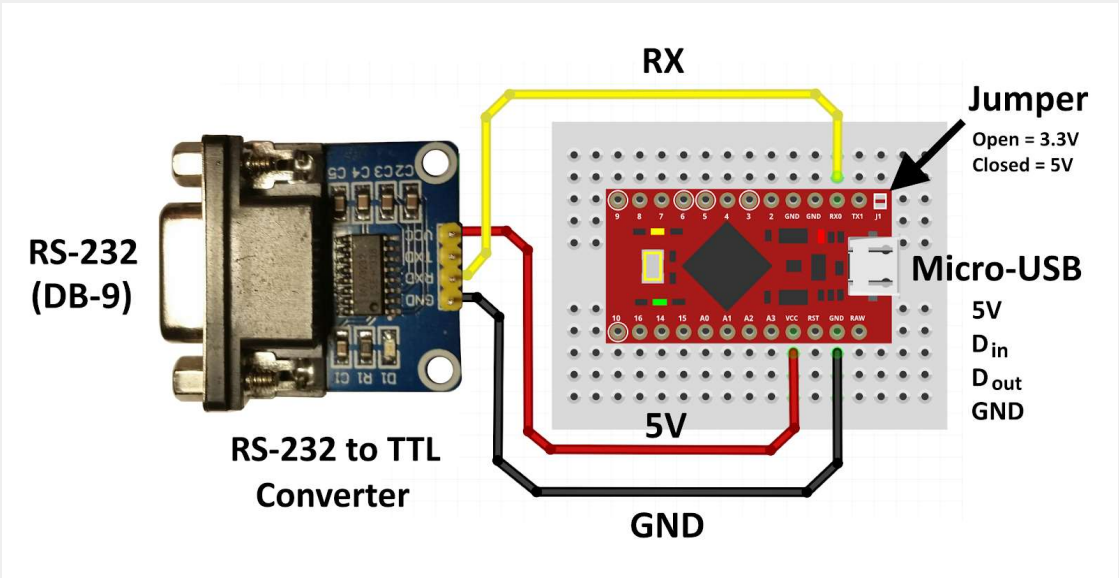
The objective for Prototype 1 was to complete a simple functional prototype and successfully test this under controlled conditions in a real world Quick Service Restaurant.

3.1.2 Components

Components ready to be used in Prototype 1 (Figure 5)
Some of the components ready to be used in the making of Prototype 1.

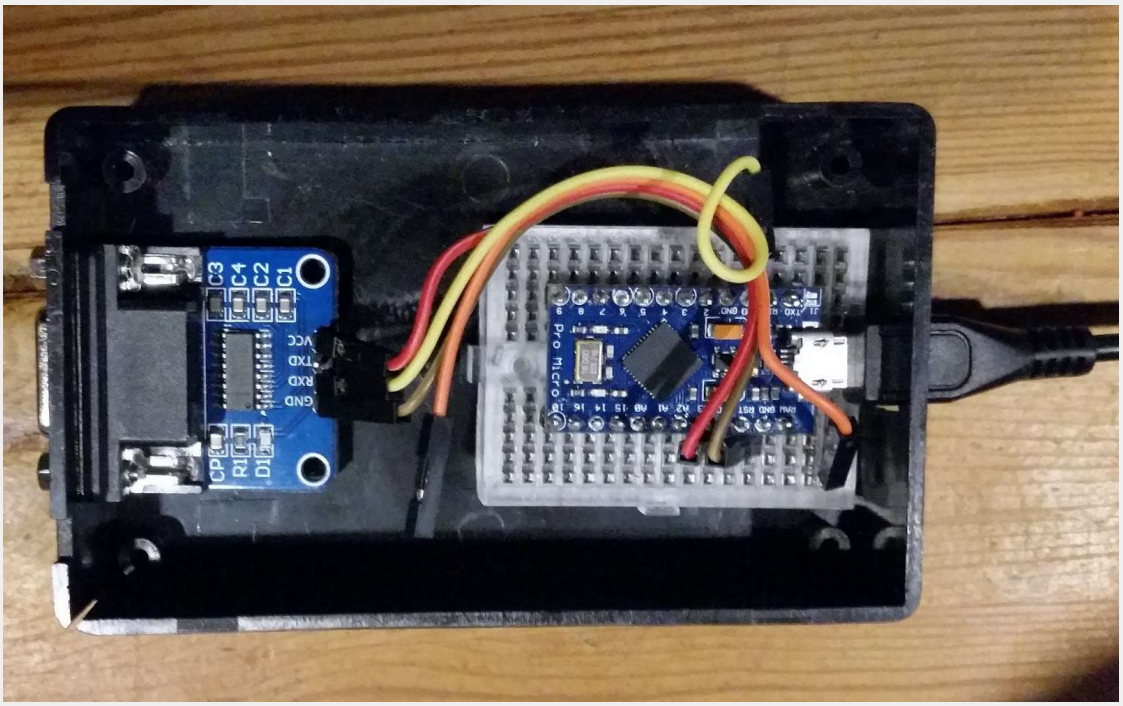
Part Description	Quantity
Enclosure	1
Mini Breadboard	1
Arduino Pro Micro (5V / 16MHz)	1
RS-232 to TTL Converter	1
Jumper connectors	4
RS-232 Splitter cable	1
DB9 Gender changer	3
USB Cable (USB-A to Micro-USB)	1

3.1.3 Breadboard Circuit



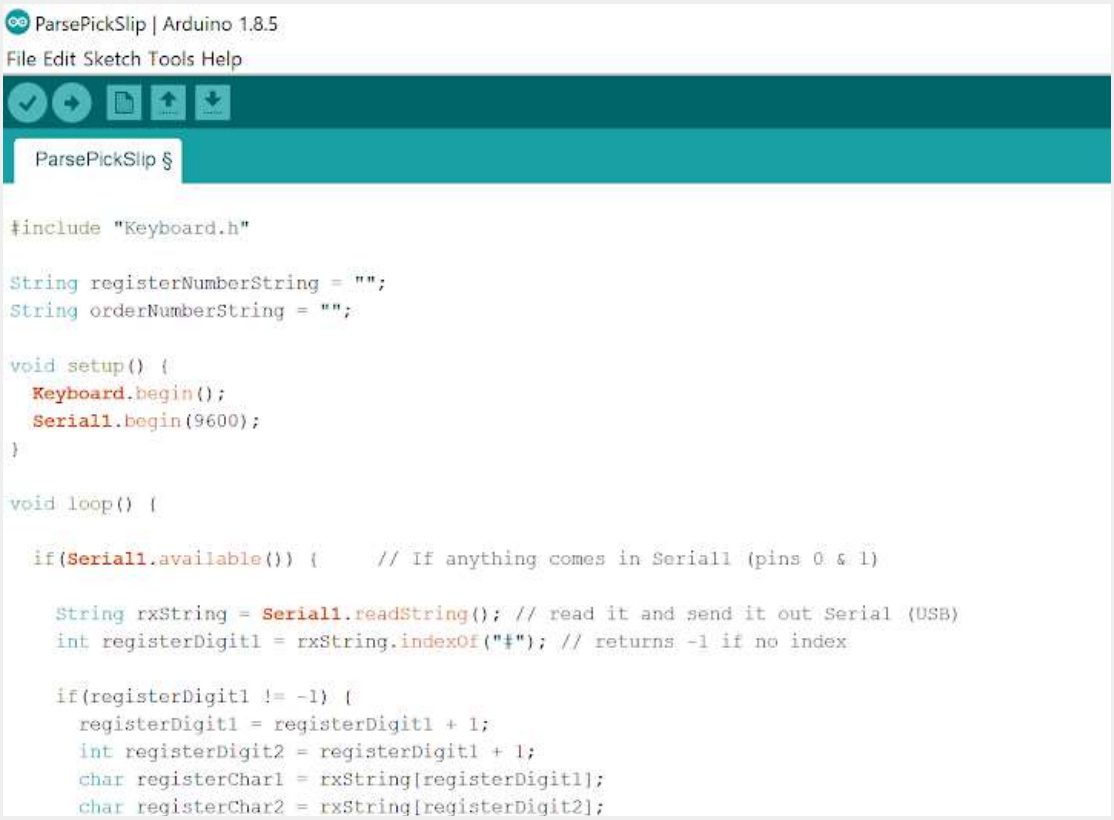
Prototype 1 Breadboard Circuit (Figure 6)
Transmit (TX) is not used because we are not sending data to RS-232.

3.1.4 Fabrication



Prototype 1 in Enclosure (Figure 7)
(The Orange wire is disconnected)

3.1.5 Microcontroller Program




```
#include "Keyboard.h"

String registerNumberString = "";
String orderNumberString = "";

void setup() {
  Keyboard.begin();
  Serial1.begin(9600);
}

void loop() {

  if(Serial1.available()) {     // If anything comes in Serial1 (pins 0 & 1)

    String rxString = Serial1.readString(); // read it and send it out Serial (USB)
    int registerDigit1 = rxString.indexOf("#"); // returns -1 if no index

    if(registerDigit1 != -1) {
      registerDigit1 = registerDigit1 + 1;
      int registerDigit2 = registerDigit1 + 1;
      char registerChar1 = rxString[registerDigit1];
      char registerChar2 = rxString[registerDigit2];
```

Microcontroller Program for Prototype 1 (Figure 8)
(refer Appendix A for full source)

3.1.6 In-House Testing

Kitchen Printer Firmware Configuration Report (Figure 9)
A Firmware Configuration Report was obtained from the kitchen printer in the Restaurant to enable testing to take place under the same conditions.

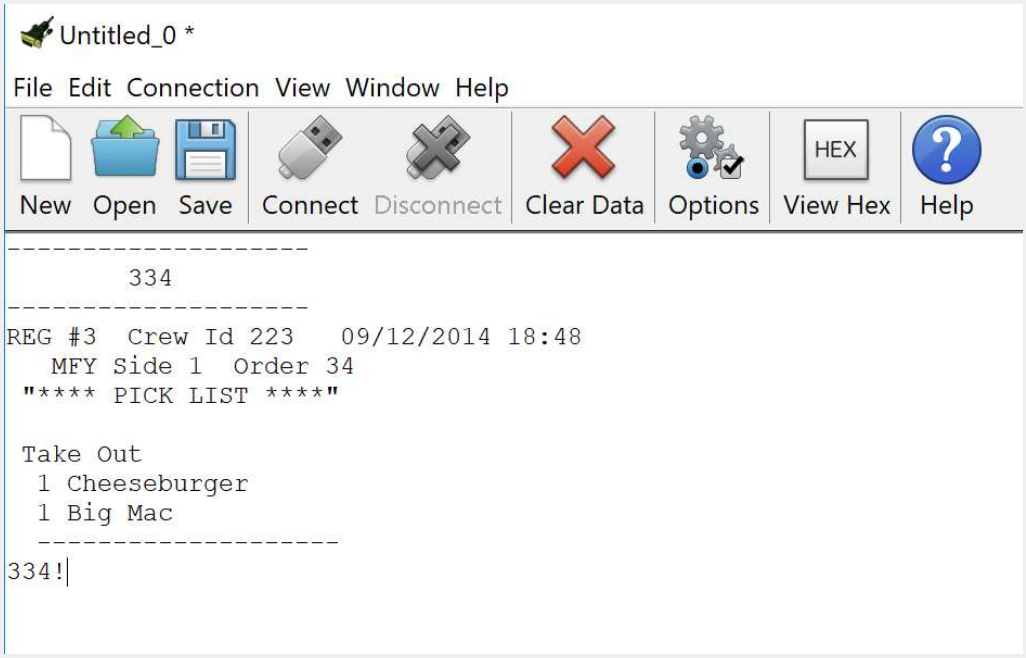
Kitchen Printer Sample Pick List (Figure 10)
Sample Pick Lists were obtained from the Restaurant.
These were helpful for generating test data suitable for testing purposes.

Testing Software

CoolTerm v1.5 was the communications utility software application selected for use with in-house testing of the prototype devices. Other applications considered include Putty and RealTerm. Either of these would also be ideal. The user friendly Graphical User Interface (GUI) of CoolTerm contributed to enabling repetitive testing to be performed faster.

Testing Procedure

Test data was sent from a PC to the device via a USB to RS-232 adapter. The microcontroller then parsed the received data and extracted the Order Number which was then promptly output through the micro-USB and back into the test computer via a different USB port. A trailing '!' character is appended to the Order Number prior to being transmitted and this is used by the receiving system as a device identifier which enables determination of whether the input data has been received from our new device or instead from a barcode scanner. This information can be useful in remote diagnostic situations.



Pick List Test Data Example (Figure 11)
Example of Pick List Test Data in use with testing a prototype device.
The last line shows the Order Number received back from the device.
The trailing '!' character is appended as our device identifier.

3.1.7 Real World Testing

Driverless Operation

Since when interfaced with the host PC the prototype device is simply doing the equivalent of emulating a keyboard the device does not receive any data back. No driver is required for the device to operate in this way and so any device driver that is either preinstalled or installed on the first connection can be simply disabled in Windows Device Manager in order to test with driverless operation.

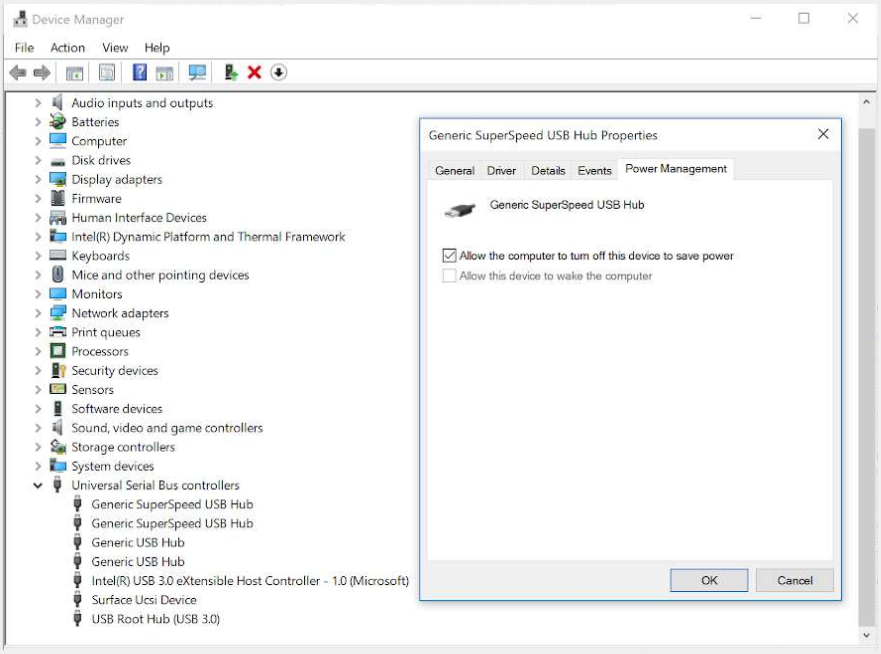
Driverless operation can be beneficial when the host PC does not have any Internet download access which otherwise may be required in order for a driver to be installed.

USB Power Saving Mode

During real world testing of Prototype 1 in a quick service restaurant one important observation was made that had not been noted during earlier testing. For the new device to continue to function the USB port on the host PC by which the device is connected must have USB Power Saving Mode disabled. This includes both in the Operating System (Device Manager in Windows) and in the PC BIOS settings.

Unless USB Power Saving Mode has been manually disabled the operating system may eventually switch off power to the connected device at which point new Order Numbers will no longer be processed or received.

This discovery also highlighted the importance of retaining a barcode scanner as a backup input method during trials with the new device as this had to be resorted to when the device unexpectedly powered down.

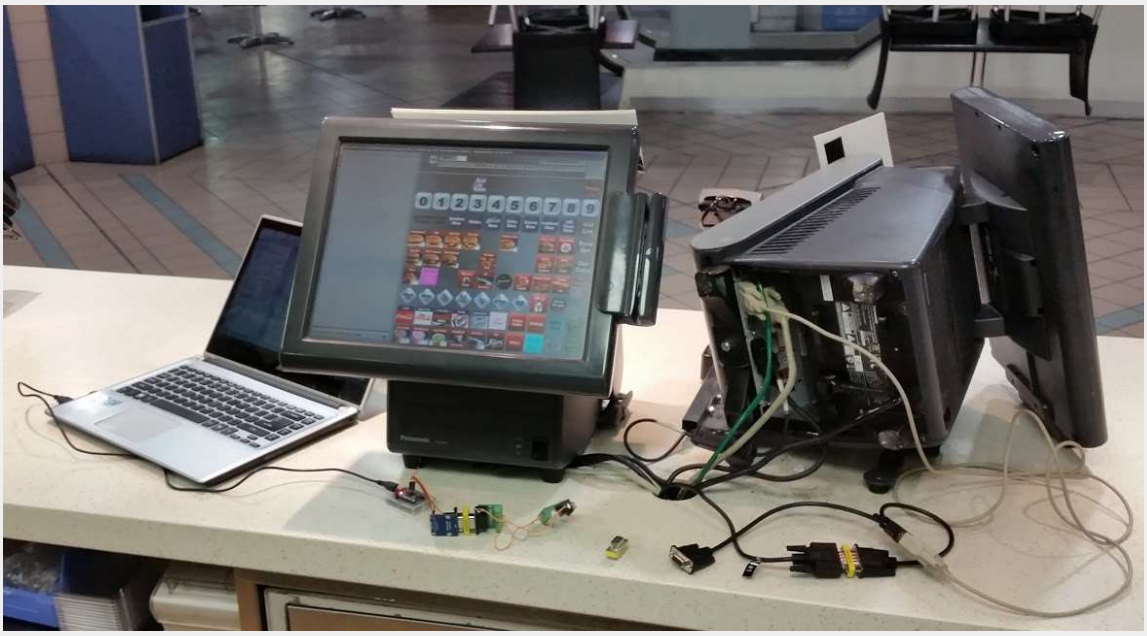


USB Power Saving Mode (Figure 12)
USB Power Saving Mode in Windows is enabled by default.
Without this being disabled the attached device will eventually power off.

Micro-USB Connector

Another issue in need of addressing was the extreme fragility of the micro-USB connector on the Arduino Pro-Micro. Even a minor pull on the USB cable could cause the connector to break irreparably and thus require a replacement Arduino.

Throughout development of the prototypes that followed a number of workarounds were experimented with including magnetic connectors as an interim solution and screw terminals as a more permanent measure. With later prototypes the vulnerability was able to be limited to during fitting of the unit into the enclosure. The magnetic connectors were able to be used to prevent additional broken units however screw terminals would prove to be a more robust solution. In future versions a USB-B female receptor soldered directly onto the PCB may prove to be a more optimal solution.



Prototype 1 Undergoing Testing (Figure 13)
Prototype 1 (without enclosure) undergoing testing with a Register POS PC.

3.2 Prototype 2

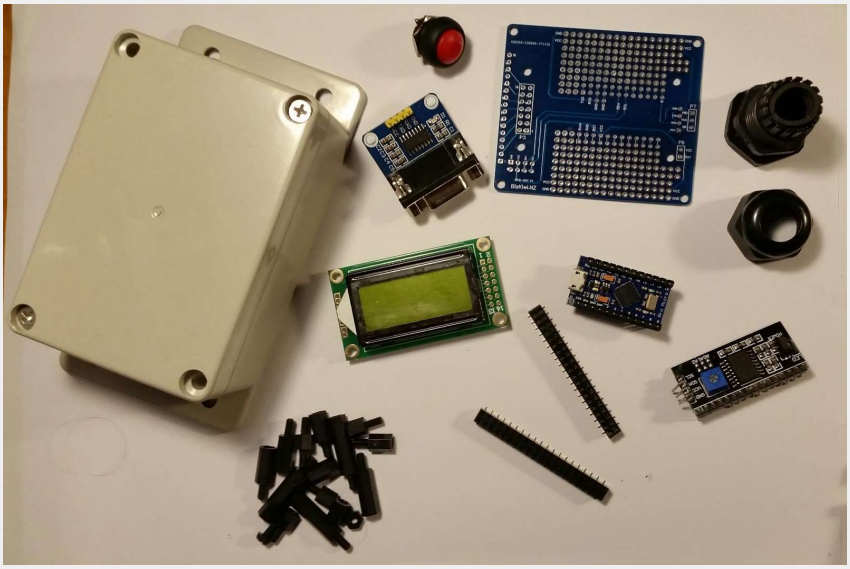
3.2.1 Functions and Features

While Prototype 1 was found to be functionally successful during trials a great deal of improvement would be needed to sustain the harsh environment of a quick service restaurant.

Goals for Prototype 2 included addition of:

- Water resistant enclosure
- Water resistant cable gland
- Water resistant Reset button
- LCD module for display of Order Numbers and Device Status

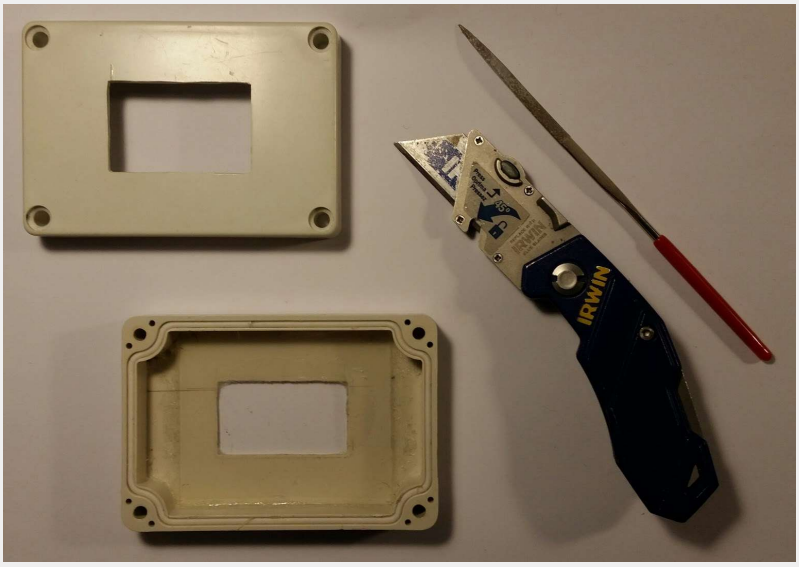
3.2.2 Components



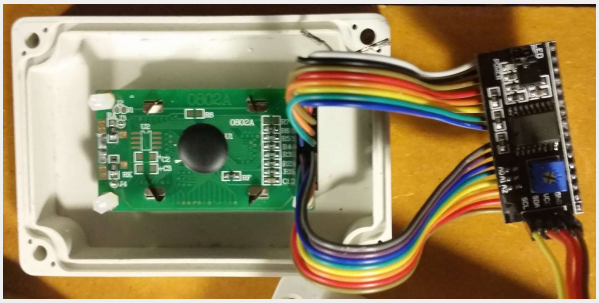
Components ready to be used in Prototype 2 (Figure 14)
Some of the components ready to be used in the making of Prototype 2.
The custom designed PCB was developed for Prototype 3.

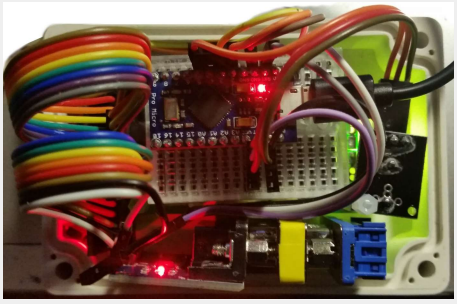
Part Description	Quantity
Enclosure	1
Mini Breadboard	1
Arduino Pro Micro (5V / 16MHz)	1
RS-232 to TTL Converter	1
Jumper wire connectors	26
RS-232 Splitter cable	1
DB9 Gender changer	3
USB Cable (USB-A to Micro-USB)	1
Reset Button (water resistant)	1
Transparent Acrylic Panel 75x60x3mm	1
PG-16 PVC Cable Gland	1
08x02 5V LCD Module (yellow backlight)	1
I2C Converter Module	1
Heatshrink 10cm (transparent)	1

3.2.3 Fabrication



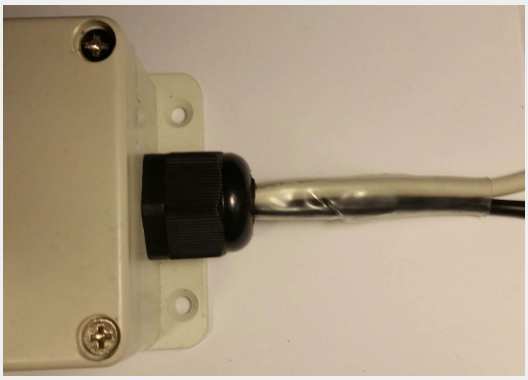
Cutting a Window for the LCD Module into Enclosure Lid (Figure 15)
A transparent acrylic panel is then glued into place for weatherproofing.





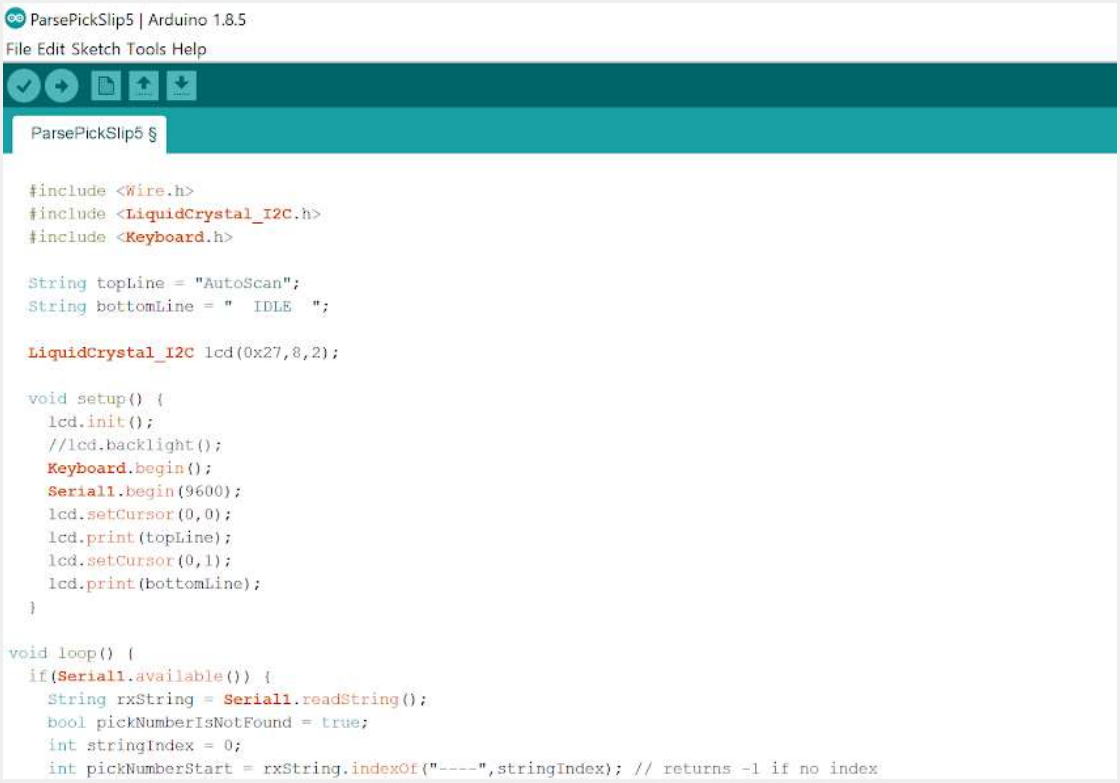
Above Left **LCD and I2C Converter Modules Placed in Enclosure. (Figure 16)**

Above Right **Prototype 2 Attached to Underside of Enclosure Lid. (Figure 17)**
There is not much unoccupied space in this prototype.



Cables inserted through a sheath of transparent heatshrink. (Figure 18 a, b)
The cables were inserted through a sheath of transparent heatshrink
to improve the seal around the cable gland entry point to the enclosure.

3.2.4 Microcontroller Program



Microcontroller Program for Prototype 2 (Figure 19)
(refer Appendix A for full source code)

3.2.5 In-House Testing

Micro USB Connector

The fragile micro-USB connector was hoped to no longer be an issue with Prototype 2 because the clamping effect of the cable gland was expected to protect the connector from movement. However, the connector was found to be at its most vulnerable during the process of installing the assembled unit inside of the enclosure and during this phase the connectors of several Pro Micros were broken irreparably.

Prototype 2 with LCD (Figure 20)
Prototype 2 with LCD indicating Order 102 has been processed.

3.2.6 Real World Testing

In-house testing of Prototype 2 had highlighted the need for design of a new PCB adapter board to eliminate the cluttered wires that had been introduced since the previous version. Prototype 2 was therefore not tested in the restaurant environment and instead work was moved directly to the development of Prototype 3.

Protective Optical Isolator (Figure 21)
A protective Optical Isolator used inline between the host PC and the splitter.

3.3 Prototype 3

3.3.1 Functions and Features

Prototype 3 Goals

Prototype 2 was functionally successful however it also highlighted some important issues to resolve. The enclosure space was too crowded and this was largely due to there being so many connection wires.

Goals for Prototype 3:

- Replacement of the mini breadboard (and the wiring introduced in Prototype 2) with a single PCB Adapter Board (PCB V1).
- Addition of protoboard space on the PCB where if needed additional components can be soldered in future device variants based around this board.

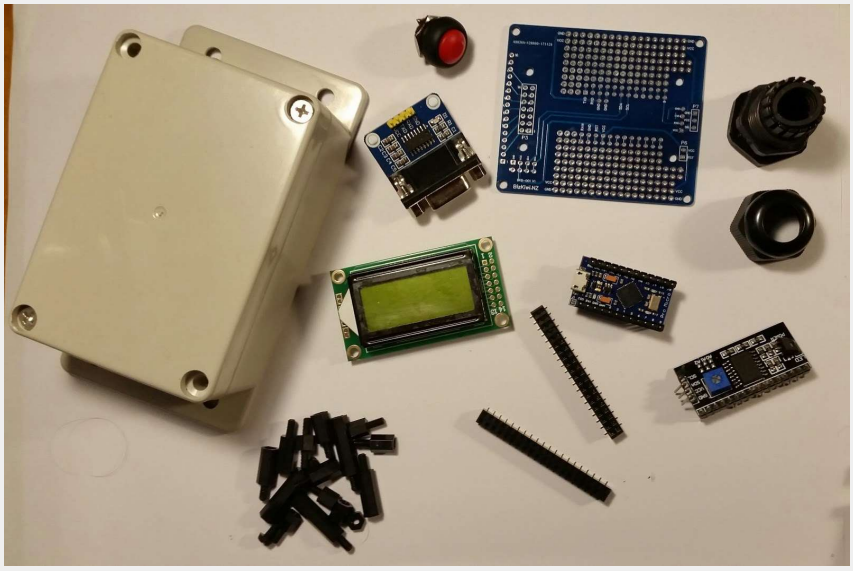
PCB Revision (PCB V1 R1)

Given the timespan from PCB order placement for manufacturing right through to receipt of delivery was only 7 days this quick turnaround would enable a revision to the PCB to be made to be manufactured in a second batch.

Changes in PCB V1 R1:

- Two minor corrections to the track layout.
- Enlargement of the PCB footprint for better use of the available enclosure space.
- The headers for the Pro Micro were moved further toward the reset button. Part of the reasoning for this was to provide more space for an inline magnetic connector to be inserted between the Pro Micro and the incoming micro USB cable. This magnetic connector was one of several ongoing attempts to protect the Pro Micro connector from damage caused by cable movement during the installation process.
- Modifications to the PCB layout to enable enlargement of the protoboard area.

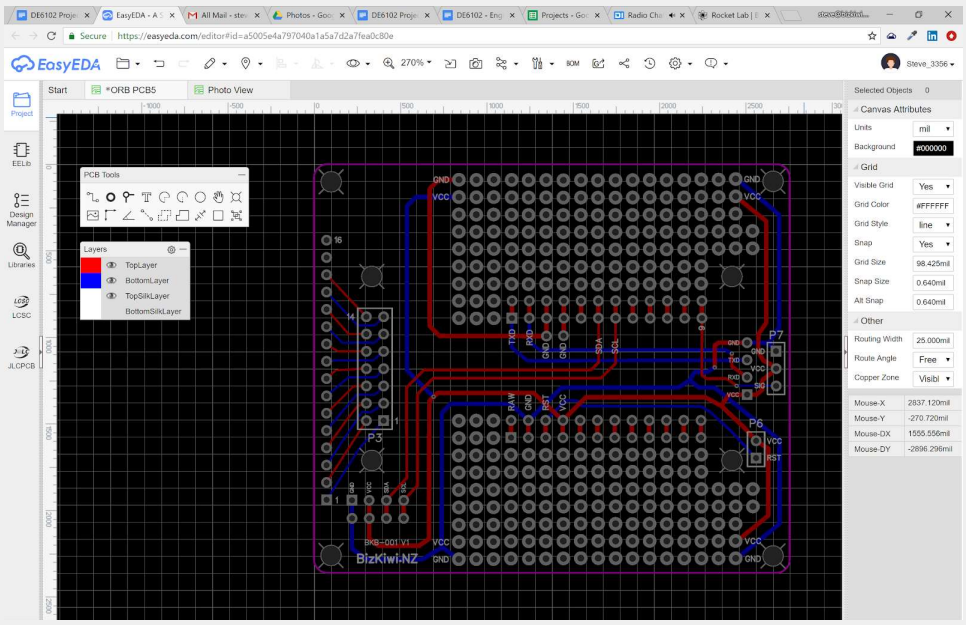
3.3.2 Components



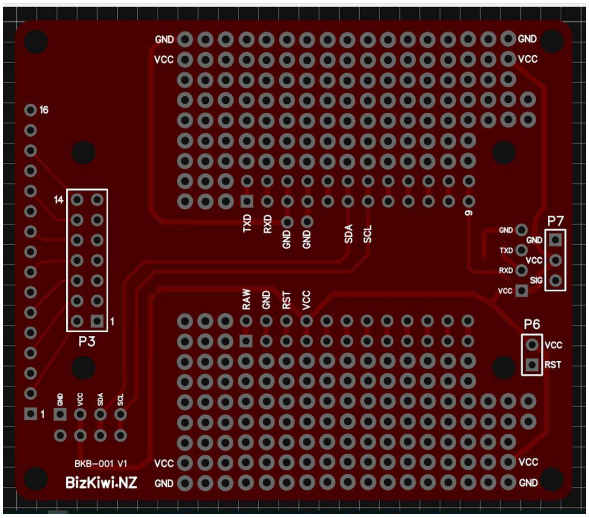
Components ready to be used in Prototype 3 (Figure 22)
Some of the components ready to be used in the making of Prototype 3.

Part Description	Quantity
Enclosure	1
Arduino Pro Micro (5V / 16MHz)	1
RS-232 to TTL Converter	1
RS-232 Splitter cable	1
DB9 Gender changer	3
USB Cable (USB-A to Micro-USB)	1
Reset Button (water resistant)	1
Transparent Acrylic Panel 75x60x3mm	1
PG-16 PVC Cable Gland	1
08x02 5V LCD Module (yellow backlight)	1
I2C Converter Module	1
Heatshrink 10cm (transparent)	1
Black Card 70x60mm (with window for LCD)	1
Silica Gel packets (for moisture absorption)	2
Custom Adapter PCB (to attach modules to)	1
Female Pin Header TH 2x7 pin	1
Female Pin Header TH 2x12 pin	2
Male Pin Header TH 4 pin	1
Screw Terminal 2.54mm 2 pin	3

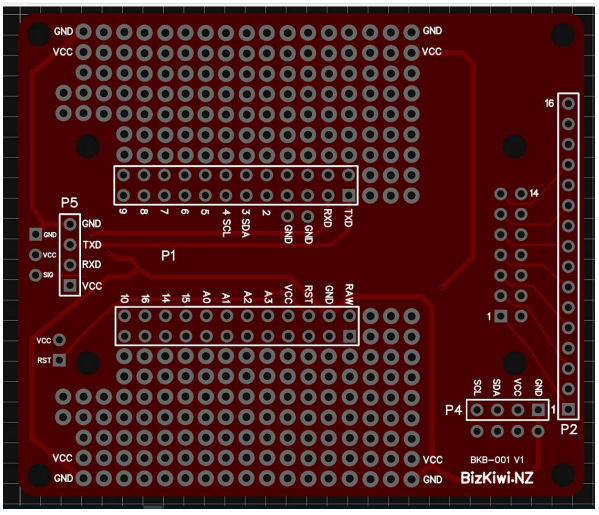
3.3.3 PCB V1



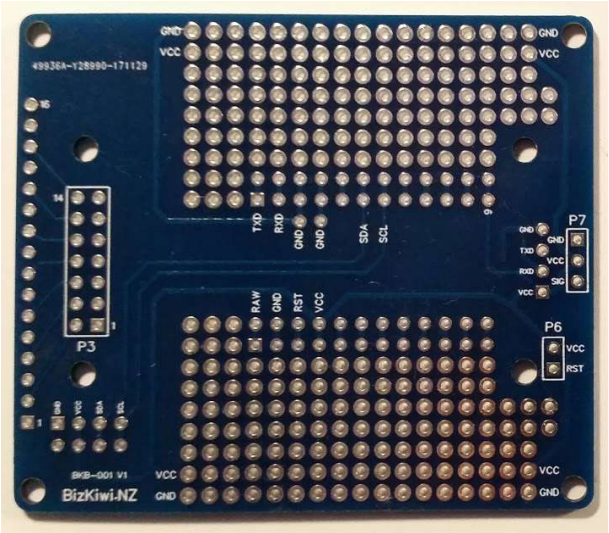
PCB Adapter Board V1 in EasyEDA (Figure 23)
EasyEDA features ordering PCB manufacturing via JLCPCB (jlcpcb.com)

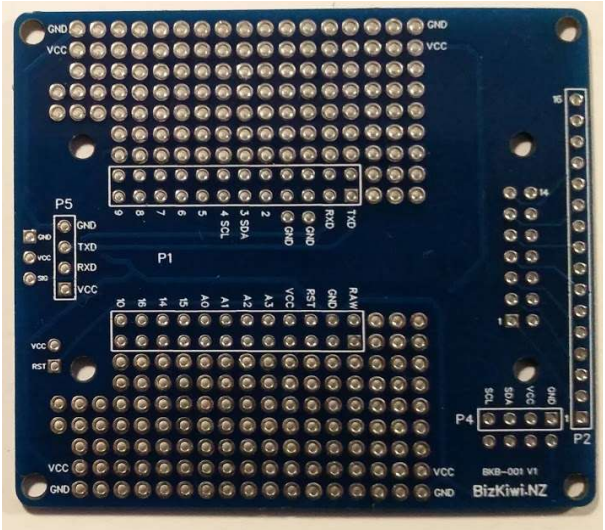


ABOVE and ABOVE RIGHT (Figure 24 a, b):
PCB V1 as rendered from within EasyEDA

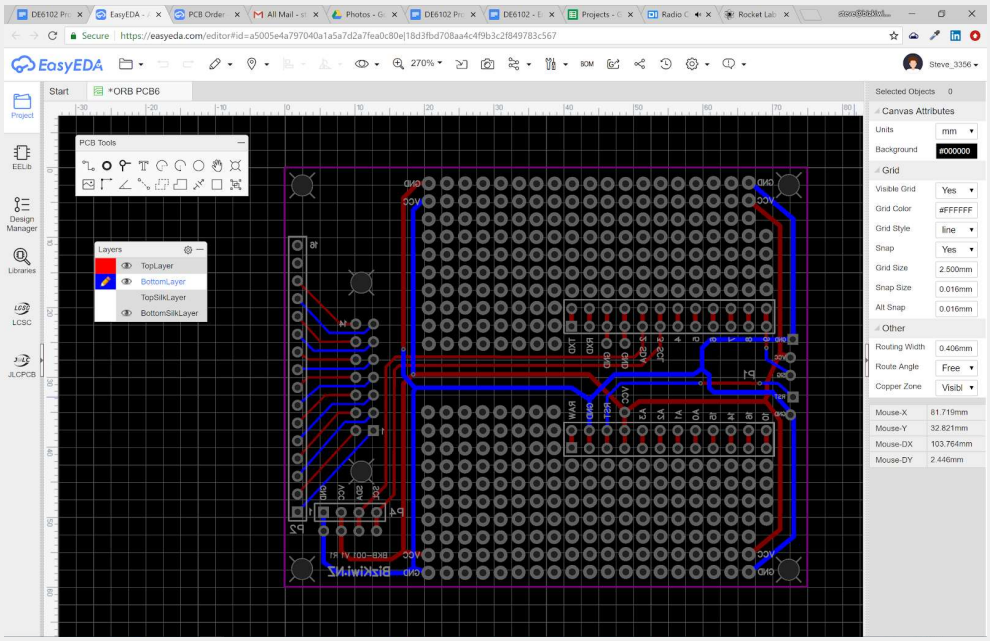


BELOW and BELOW LEFT (Figure 25 a, b):
PCB V1 as received from Manufacturer.

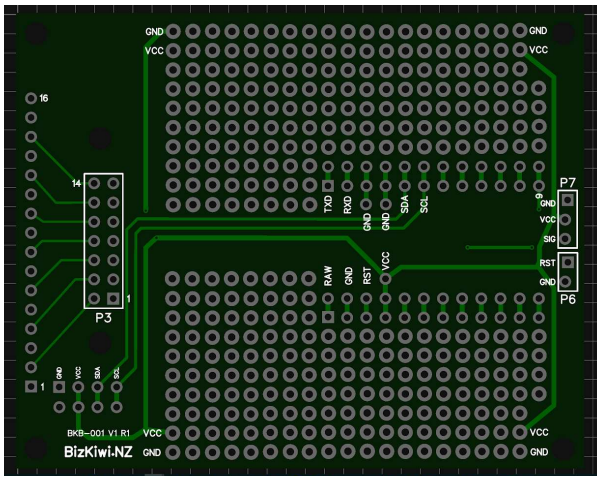




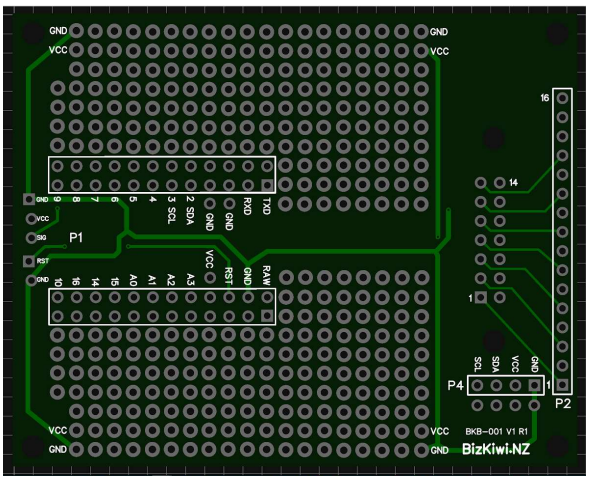
3.3.4 PCB V1 Revision 1



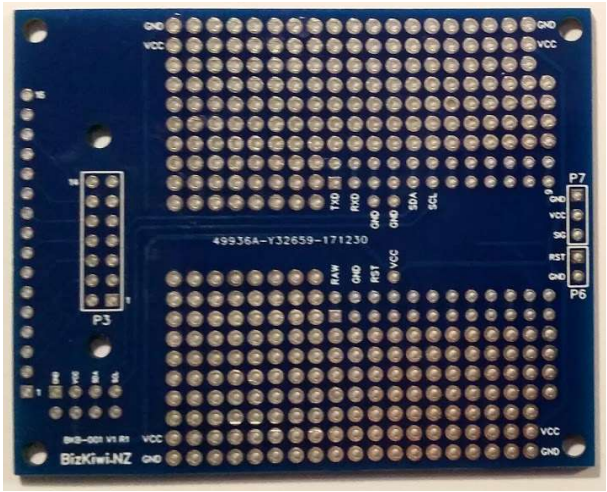
PCB V1 R1 with enlarged protoboard design in EasyEDA (Figure 26)
Order placement to delivery is 7 days making multiple revisions practical.

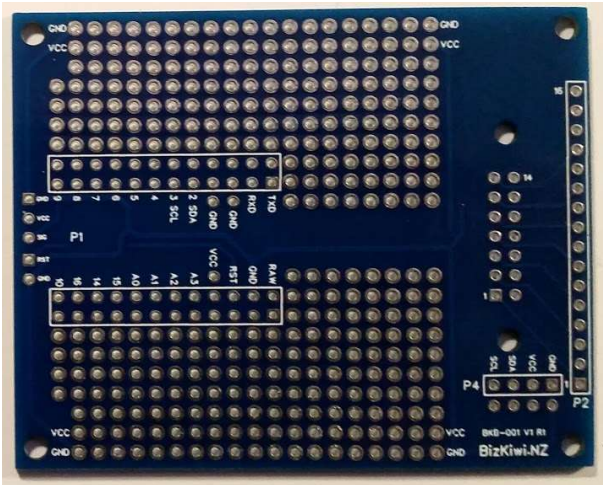


ABOVE and ABOVE RIGHT (Figure 27 a, b):
PCB V1 R1 as rendered from within EasyEDA

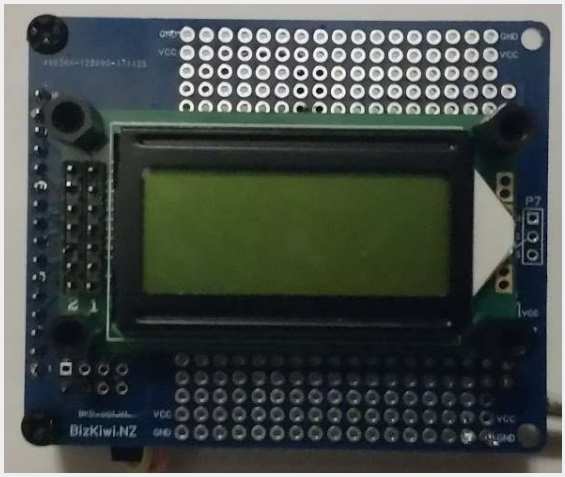


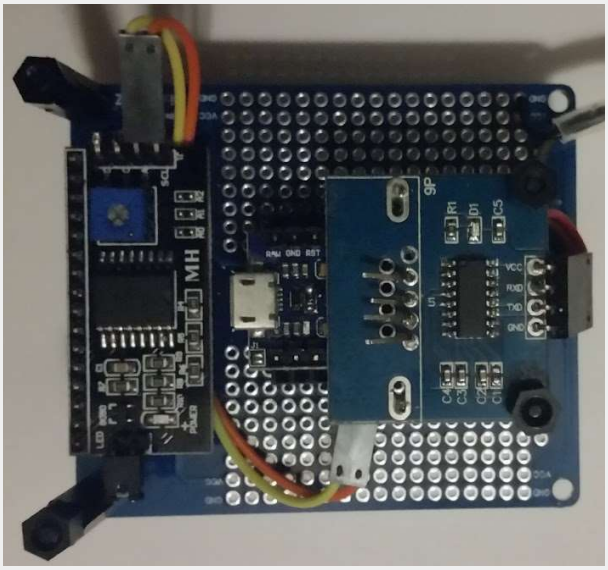
BELOW and BELOW LEFT (Figure 28 a, b):
PCB V1 R1 as received from Manufacturer.



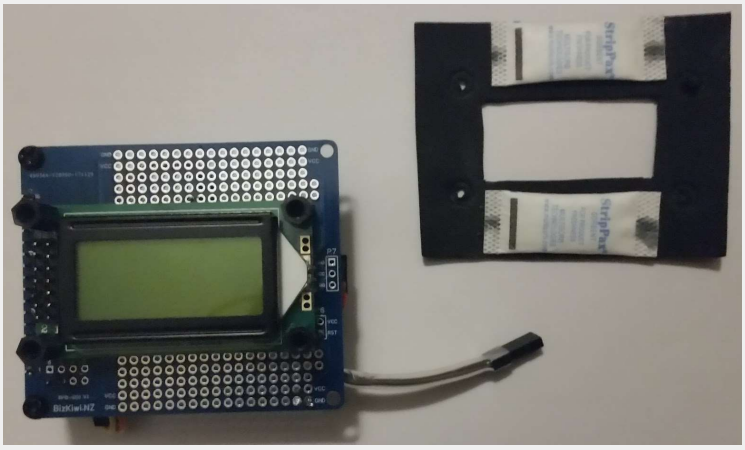


3.3.5 Assembled Unit

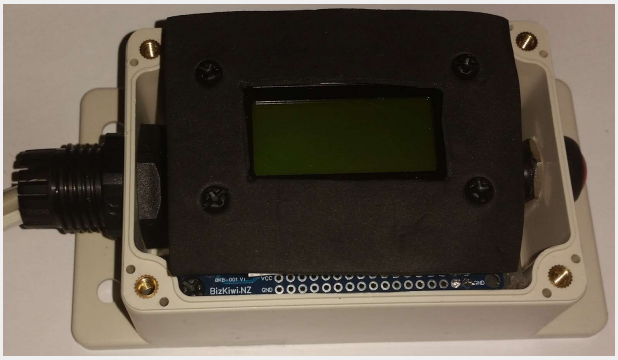




PCB V1 as Assembled (Figure 29 a, b)
Here is an assembled V1 PCB. The yellow and orange wires will be removed in Revision 1.
The nylon spacers will help with correct positioning of the unit inside of the enclosure.



LCD Module Frame (Figure 30)
To frame the LCD module neatly a window was cut in a rectangular piece of black card.
Two silica gel packs glued to the underside will absorb moisture once in the enclosure.



LCD Frame attached to Unit (Figure 31)
LCD frame attached to unit assembly and
positioned inside of the prepared enclosure.

LCD positioned with lid closed (Figure 32)
Enclosure screws are fastened tight.
The outer cable gland housing is screwed on.

3.3.6 Microcontroller Program

Prototype 3 succeeded in saving a great deal of enclosure space by replacing most of the wiring (from Prototype 2) with equivalent tracks on a simple PCB adapter board.

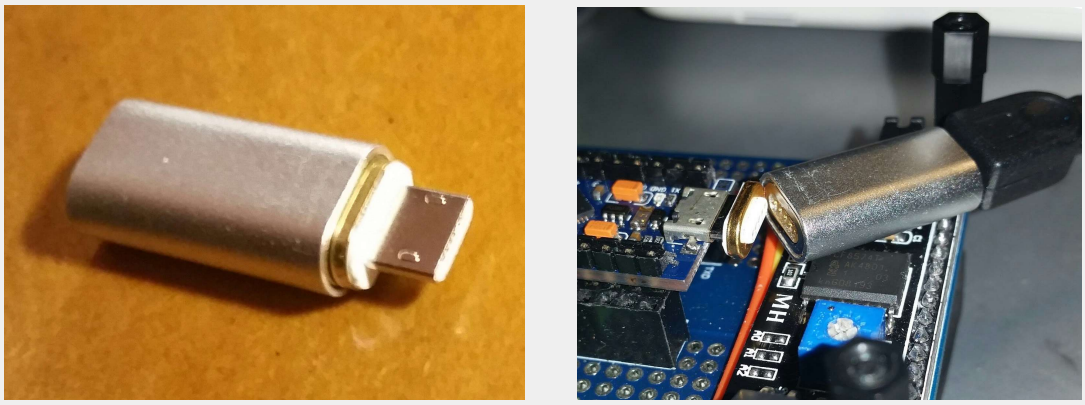
Yet functionally Prototype 3 remains identical to Prototype 2 and therefore no new changes were required to be introduced to the microcontroller software. Appendix A shows the full source code.

3.3.7 In-House Testing

Magnetic Connector

With Prototype 3 a further attempt was made to find an interim solution for the problematic issues with the micro USB connector. A magnetic connector adapter was sourced and the layout for PCB V1 R1 had been modified to allow one of these to be inserted inline between the Pro Micro and the incoming micro USB cable.

This did in fact turn out to provide successful results during the installation phase where the micro-USB connector was at its most vulnerable. Unfortunately, during the testing that followed the cable gland was found to be unable to provide enough clamping effect to prevent potential for disconnection of the magnetic connector when the cable external to the enclosure was flexed.



Micro-USB magnetic connector adapter (Figure 33 a, b)
A magnetic connector adapter was placed inline between the incoming micro-USB connector and the Pro Micro female micro-USB receptor. With pressure applied and released the connector will snap back into place.

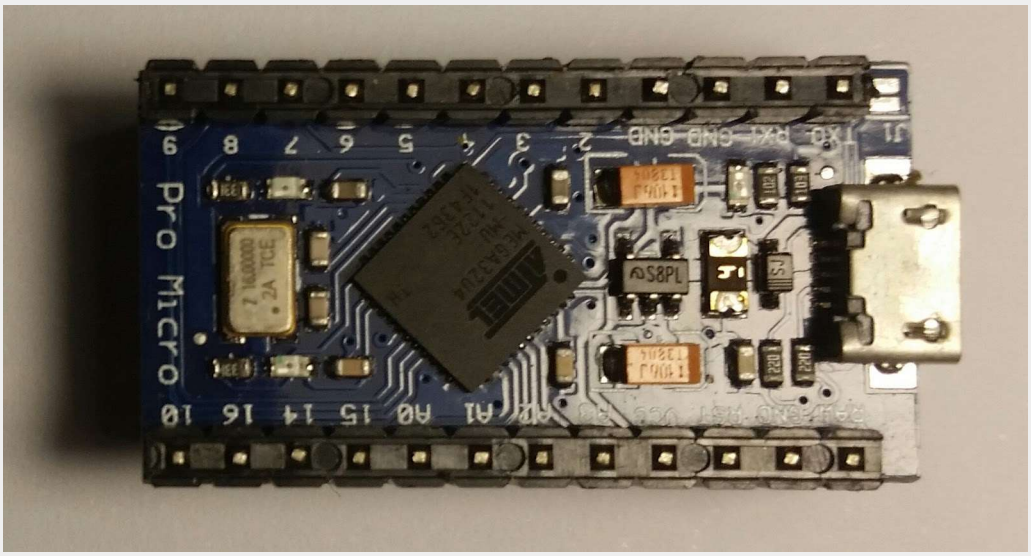
Screw Terminals

As a measure to resolve the fragility issues of the Pro Micro micro-USB connector sufficiently enough to move to real world testing the USB cable was cut and wired directly to screw terminals soldered to the PCB. The cut off cable piece with the micro-USB connector had the cut wires also screwed into the screw terminals after the micro-USB connector was connected to the Pro-Micro. The screw terminals thus acted as a pressure point and an anchor protecting the fragile micro-USB connector on the Pro Micro from potentially damaging strain.

3.3.8 Real World Testing

The restaurant was provided with two working versions of Prototype 3 complete with water resistant enclosures.

Two Tested Prototype 3 Units (Figure 34)
The restaurant was provided with two tested units of Prototype 3 each complete with water resistant enclosure, cable gland and reset button.



Pro Micro Arduino Microcontroller (Figure 35)
This Pro Micro Arduino was initially believed to be the 5V 16MHz type.
To operate at 5V the jumper J1 (at top right corner) should be closed otherwise the Arduino operate outside of its designed operating limits.
This may explain why early prototypes failed after a few days.

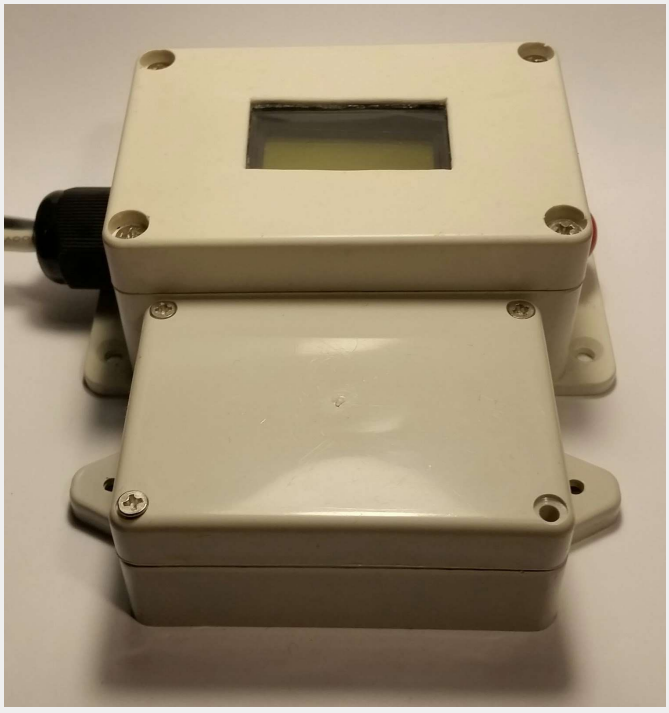
3.4 Additional Prototypes

3.4.1 Prototype 4

At the submission deadline for this Report planning work was already underway for Prototype 4. As with earlier prototypes, Prototype 4 would build on what was learned from previous versions. A number of new features would be added with a focus on miniaturisation.

Goals for Prototype 4:

- Replace 24 pin Pro Micro (1 serial port) with 28 pin Teensy 3.2 (3 serial ports)
- 3 Serial ports: replace RS-232 to TTL module with MAX3232 chip soldered to PCB.
- Optocoupler isolation for each serial port.
- Isolated DC-DC converter for each optocoupler circuit.
- Investigate use of MAX3232 series IC's with on-board optocouplers.
- On board resetting current limiting fuse
- Surge protection circuit
- Surface mounted Piezoelectric Buzzer or Speaker for audio alert sounds
- Integration of components from I2C LCD module with new PCB
- Realtime Clock (RTC) with battery cell
- EEPROM for non-volatile data storage.
- USB-B female receptor soldered to PCB
- All surface mounted components wherever possible
- New 12x02 character slimline LCD
- Smaller enclosure

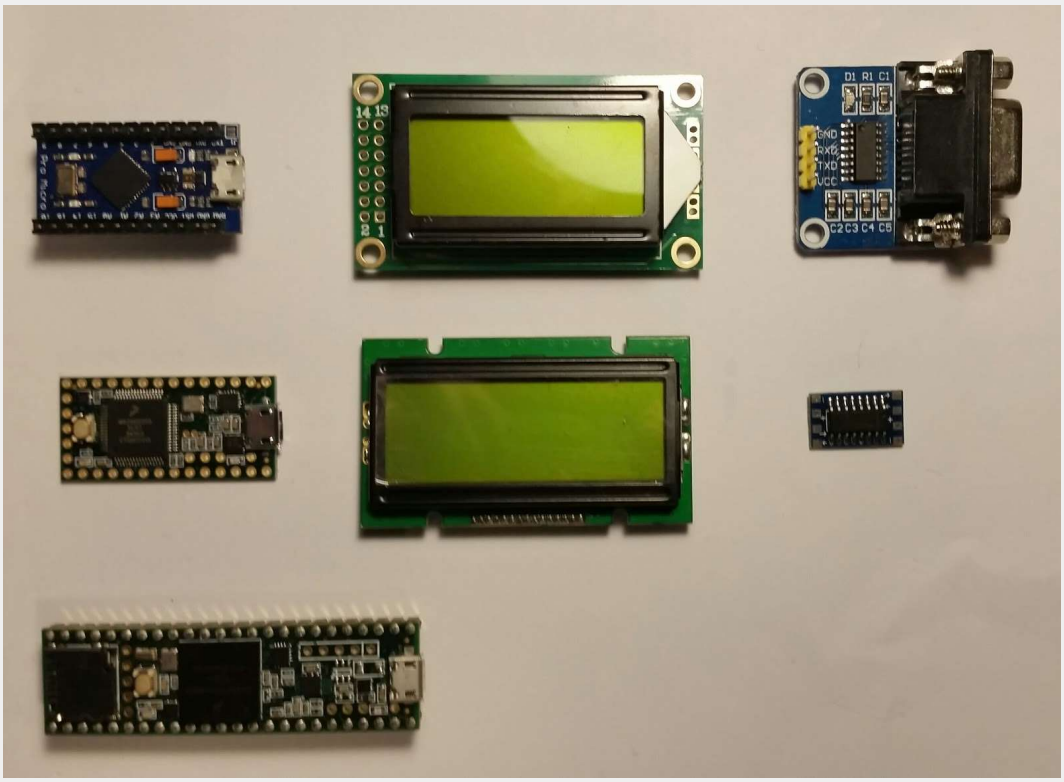


Prototype 4 Enclosure next to Prototype 3 Enclosure (Figure 36)
Prototype 4 would use the smaller enclosure in the foreground.
A completed Prototype 3 device is shown in the background.

3.4.2 Features for Future Prototypes

	Prototype	Application / Purpose / Benefit
Protection Circuits		
- Resettable Current Limiting Fuse	4	Safety, protection of device and external equipment.
- Circuit Isolation on PCB	4	Safety, protection of device and external equipment.
- Voltage Regulator on PCB	4	Safety, protection of device and external equipment.
- Surge and Overcurrent Protection	4	Safety, protection of device and external equipment.
- Optical or Magnetic Isolators	4	Isolation of RS-232 data lines.
- Isolated DC-DC Converters	4	Supply of voltage to Optical or Magnetic Isolators
Microcontroller Options		
- AtMega32u4 SMD and oscillator	?	Microcontroller can be integrated with the PCB.
- or Teensy LC / 3.2 (via 2x14pin headers)	4	Enables connection of up to 3 serial devices.
- Realtime Clock (RTC) with battery cell	4	Time tracking even without external power source.
- EEPROM IC	4	Non-volatile data storage.
Connectivity Options		
- Power-over-Ethernet (PoE)	?	Single cable for power and data connection to server.
- WiFi connectivity (ESP32 or ESP8266)	?	Connection to server.
- Bluetooth connectivity (ESP32)	?	Emulation of bluetooth keyboard.
- 3 x RS-232 Channels	4	Kitchen needs to use 2 printers at peak times.
- 6 x RS-232 Channels	5	Enables monitoring of additional printers.
- Network Time Server (NTP) connectivity	?	Synchronisation with Atomic Clock.
Instrumentation and Control		
- 12x2 character slimline LCD	4	- Displays 24 characters of text. (Prototypes 2 and 3 displayed 16 chars). - More compact requiring less enclosure space.
- SMD Piezoelectric Transducer or Speaker	4	Buzzer or MIDI / Audio event alert sounds
- Solid State Relay (12/24V?)	5	e.g. switch on connected flashing lights.
Miniaturisation		
- Surface Mounted (SMD) components	4	Miniaturisation, lower manufacturing cost.
- Single Board PCB	4	All components mounted on a single PCB
- Smaller Enclosure	4	Space saving enclosure.

3.4.3 Components for Future Prototypes



Components for Future Prototypes (Figure 37)

	Microcontroller	LCD Module	RS-232 Interface
Top Row used in Prototypes 2 / 3	Pro Micro (1 serial port)	08x02 LCD (16 chars)	RS-232 to TTL Converter
Second Row for use in Prototype 4	Teensy 3.2 (3 serial ports)	12x02 LCD (24 chars)	MAX3232 IC Module
Front for use in Future Version?	Teensy 3.5 (6 serial ports)		

4 Conclusion

4.1 Project Outcomes

The developed system has been successfully demonstrated but is not yet ready for use outside of controlled conditions within a commercial environment. Compliance and certifications are needed before a device can be considered ready to commercialise.

4.2 Compliance and Certifications

In an ideal world a commercially acceptable product would feature compliance with:

- RoHS2
- CE-Mark
- FCC
- UL Certification

4.3 Summary of Learning

A full breakdown of modern tool usage in this project is included in Appendix C.

4.4 Recommendations

None of the prototypes built so far are ready to be used as commercial products and further development is needed in order to reach this point. Each prototype built has built upon what was learned from the previous one. The as yet unbuilt Prototype 4 would be another step forward yet it is fully expected that further development stages will be required in order to progress sufficiently to arrive at a commercially acceptable product.

Project Poster Displayed at Project Presentations (Figure 38).
(Photo Credit: Huan Zhuang - Friday 15 June 2018)

List of Figures

All photographs, diagrams and screenshots are by the author unless otherwise indicated.

References

National Instruments (28/12/2017). *How to do a serial loopback test.* Retrieved from http://www.ni.com/tutorial/3450/en/

Dallas Semiconductor (n.d.). *Application note 83 - Fundamentals of RS-232 serial communications.* Retrieved from http://ecee.colorado.edu/~mcclurel/dan83.pdf

Spark Fun Electronics (n.d.). *Pro Micro - 5v / 16mhz.* Retrieved from https://www.sparkfun.com/products/12640

Spark Fun Electronics (n.d.). *Pro Micro schematic v13b.* Retrieved from https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/Pro_Micro_v13b.pdf

Deskthority Wiki (n.d.) *Arduino Pro Micro.* Retrieved from https://deskthority.net/wiki/Arduino_Pro_Micro

Appendix A

Firmware for AutoScan V1 (as used in Prototype 1)

```
1
2 // AutoScanV1 by Stephen Julian
3 // Published with MIT licence 2018
4
5 #include <Keyboard.h>
6
7 String registerNumberString = "";
8 String orderNumberString = "";
9
10 void setup() {
11   Keyboard.begin();
12   Serial1.begin(9600);
13 }
14
15 void loop() {
16   if(Serial1.available()) {     // If anything comes in Serial1 (pins 0 & 1)
17     String rxString = Serial1.readString(); // read it and send it out Serial (USB)
18     int registerDigit1 = rxString.indexOf("#"); // returns -1 if no index
19     if(registerDigit1 != -1) {
20       registerDigit1 = registerDigit1 + 1;
21       int registerDigit2 = registerDigit1 + 1;
22       char registerChar1 = rxString[registerDigit1];
23       char registerChar2 = rxString[registerDigit2];
24       if(isDigit(registerChar2)) {
25         registerNumberString = rxString.substring(registerDigit1,registerDigit2+1);
26       } else if(isDigit(registerChar1)) {
27         registerNumberString = rxString.substring(registerDigit1,registerDigit1+1);
28       }
29     }
30     if(registerNumberString != "") {
31       int orderDigit1 = rxString.indexOf("Order");
32       if(orderDigit1 != -1) {
33         orderDigit1 = orderDigit1 + 5;
34         char orderChar1 = rxString[orderDigit1];
35         bool charIsDigit = isDigit(orderChar1);
36         while(!charIsDigit) {
37           orderDigit1++;
38           orderChar1 = rxString[orderDigit1];
39           charIsDigit = isDigit(orderChar1);
40         }
41         char orderChar2 = rxString[orderDigit1+1];
42         charIsDigit = isDigit(orderChar2);
43         if(charIsDigit) {
44           orderNumberString = rxString.substring(orderDigit1,orderDigit1+2);
45         } else {
46           orderNumberString = rxString.substring(orderDigit1,orderDigit1+1);
47         }
48       }
49     }
50     if((registerNumberString != "") && (orderNumberString != "")) {
51       String outputString = registerNumberString;
52       outputString.concat(orderNumberString);
53       registerNumberString = "";
54       orderNumberString = "";
55       // delay(3000); // delay for use only when testing
56       Keyboard.print(outputString);
57     }
58   }
59 }
```

Firmware for AutoScan V2 (as used in Prototypes 2 and 3)

```
// AutoScanV2 by Stephen Julian
// Published with MIT licence 2018

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keyboard.h>

String topLine = "         ";
String bottomLine = "         ";
LiquidCrystal_I2C lcd(0x27,8,2);

void setup() {
  lcd.init();
  //lcd.backlight(); // this line only relevant for 16 pin LCD modules
  Keyboard.begin();
  Serial1.begin(9600);
}

void loop() {
  if(Serial1.available()) {
    String rxString = Serial1.readString();
    bool pickNumberIsNotFound = true;
    int stringIndex = 0;
    int pickNumberStart = rxString.indexOf("----",stringIndex); // returns -1 if no index
    if(pickNumberStart != -1) {
        while(pickNumberIsNotFound) {
        pickNumberStart = pickNumberStart + 1;
        int pickNumberDigit1 = pickNumberStart;
          char pickNumberChar1 = rxString[pickNumberDigit1];
        if(isDigit(pickNumberChar1)) {
          int pickNumberDigit2 = pickNumberDigit1 + 1;
          char pickNumberChar2 = rxString[pickNumberDigit2];
          if(isDigit(pickNumberChar2)) {
            int pickNumberDigit3 = pickNumberDigit2 + 1;
            char pickNumberChar3 = rxString[pickNumberDigit3];
            if(isDigit(pickNumberChar3)) {
              int pickNumberDigit4 = pickNumberDigit3 + 1;
              char pickNumberChar4 = rxString[pickNumberDigit4];
              String outputString = "";
              outputString.concat(pickNumberChar1);
              outputString.concat(pickNumberChar2);
              outputString.concat(pickNumberChar3);
              if(isDigit(pickNumberChar4)) {
                outputString.concat(pickNumberChar4);
              }
              int numChars = outputString.length();
              if(numChars == 3) {
                topLine = "  #";
                topLine.concat(outputString);
                topLine.concat("  ");
              } else { // numChars == 4
                topLine = " #";
                topLine.concat(outputString);
                topLine.concat("  ");
              }
              if(outputString.length() < 4) {
                outputString.concat('!');
              }
              //delay(3000); // delay for use only when testing
              Keyboard.print(outputString);
              lcd.setCursor(0,0);
              lcd.print(topLine);

              lcd.setCursor(0,1);
              lcd.print(" SERVED ");
              pickNumberIsNotFound = false;
            }
          }
        }
      }
    }
  }
}
```

Appendix B

Bill of Materials (BOM) for Prototype 3

Part Description	QTY	Unit Price	Sub-Totals
Enclosure	1	4.90	4.90
Arduino Pro Micro (5V / 16MHz)	1	9.00	9.00
RS-232 to TTL Converter	1	1.00	1.00
RS-232 Splitter cable	1	5.00	5.00
DB9 Gender changer	3	1.00	3.00
USB Cable (USB-A to Micro-USB)	1	1.80	1.80
Reset Button (water resistant)	1	1.50	1.50
Transparent Acrylic Panel 75x60x3mm	1	0.50	0.50
PG-16 PVC Cable Gland	1	1.50	1.50
08x02 5V LCD Module (yellow backlight)	1	3.50	3.50
I2C Converter Module	1	1.90	1.90
Heatshrink 10cm (transparent)	1	0.30	0.30
Black Card 70x60mm (with window for LCD)	1	0.20	0.20
Silica Gel packets (for moisture absorption)	2	0.20	0.40
Custom Adapter PCB (to attach modules to)	1	3.40	3.40
Female Pin Header TH 2x7 pin	1	1.00	1.00
Female Pin Header TH 2x12 pin	2	1.80	3.60
Male Pin Header TH 4 pin	1	0.40	0.40
Screw Terminal 2.54mm 2 pin	3	0.80	2.40
		TOTAL $	**45.30**

Appendix C

Modern Tool Usage

	Application / Purpose / Benefit
Software Tools	
- Fritzing 0.9.3 beta	Circuit prototype design tool.
- Atmel Studio	Software development for AtMega32u4 microcontroller.
- Arduino IDE	Uploading of firmware to microcontroller.
- CoolTerm 1.5	Communications tool used to send test data to prototypes.
Web Services	
- EasyEDA (easyeda.com)	Electronic design of Printed Circuit Boards (PCB). Order placement for manufacture of batches of PCB's.
- Google Apps for Business	Communication, Documentation and file storage.
Hardware Tools	
- Laptop PC	Software development, PCB design, device testing, documentation.
- Smartphone	Capturing photos of device fabrication, assembly and testing.
- Multimeter	Continuity testing (e.g. of PCB's from manufacturer)
- USB to RS-232 cable	Cable adapter for interfacing host PC with prototype devices.
- Thermostatic Soldering Iron and Accessories	Soldering of prototypes.
- Cordless Drill and Accessories	Drilling of hole for cable ports / cable glands and reset buttons. Accessories included drill bits for drilling extra large holes.
- Electric Heat Gun	For heating heat shrink enclosing cables in cable gland areas.
- Hand Tools	Fabrication and/or modification of device enclosures. Tools found to be useful include: Screwdrivers, pliers, craft knife, hacksaw, files, sandpaper.
Consumables	
- Lead-Free Solder	Soldering of prototypes.
- Waterproof Adhesive	Applied around the edges of the transparent acrylic panels prior to installation inside the windows cut into the enclosure lid.
- Silicone Sealant	Applied to vulnerable parts of enclosure to increase resistance to water. A small amount was injected into the cable gland and around the reset button inside of the enclosure just prior to closing and fastening the lid.