Towards Industry 4.0: Sensorizing CNC Machine

Team Members (Student ID):

Zhao Zedong (1534436)
Muhammad Aidil Bin Azman (1549962)

Final Year Project Report

Supervisor: Dr. Win Tun Latt
Acknowledgement

In the making of project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. We would like to show our gratitude to the following SP lecturers and staff for their guidance and support to help making this project possible:

Dr Win Tun Latt, project supervisor for giving us a good guideline for assignment throughout numerous consultations. He is willing to take all his time to ensure our project is able to progress smoothly. He also source together with us for the material that is needed. He has been a great support to our group. We appreciate his endless effort and the time that he put in for our project.

Many people, especially our friends and team members itself, have made valuable comment suggestions on this project which gave us an inspiration to improve. We thank all the people for their help directly and indirectly to complete our Final Year Project.
Abstract

This report is intended to introduce to people about our final year project of sensorizing the Computer Numerical Control (CNC) machine. In this report, it shows the detail of the components, cloud platform and program we discovered and used to sensorise the machine. The report will also mention about the interface between different components and devices. It also describes the detail of connecting the sensors to the cloud platform and the machine, and installing the sensors to the machine. In the end of the report, it includes the further improvement that we can do in the future for this project.
Table of Content

1.0 Introduction .................................................................................................................. 4
  1.1 Background of Project ............................................................................................... 4
  1.2 Objective of Project ..................................................................................................... 5
  1.3 Scope of Project .......................................................................................................... 6

2.0 Research of Components .............................................................................................. 7
  2.1 Sensors .......................................................................................................................... 7
  2.2 Embedded Controller/Computer .................................................................................. 10
  2.3 Types of Interface ....................................................................................................... 11
  2.3 Cloud Platforms .......................................................................................................... 12
  2.4 Selected Components ................................................................................................. 13

3.0 Connection .................................................................................................................... 16
  3.1 Raspberry Pi .................................................................................................................. 16
  3.2 Cloud Platform and Raspberry Pi ............................................................................... 26

4.0 Installation ...................................................................................................................... 32

Problem Faced ..................................................................................................................... 43
Recommendation ................................................................................................................. 44

Appendix A .......................................................................................................................... 46
Appendix B .......................................................................................................................... 47
Appendix C .......................................................................................................................... 48

References ............................................................................................................................ 49
Introduction

1.1 Background of Project

Industry 4.0 is the fourth industrial revolution, it is a name for the current trend of automation and data exchange in manufacturing technologies. It includes cyber-physical systems, the Internet of things (IOT), cloud computing and cognitive computing. Industry 4.0 originating from Germany as part of a Governmental strategy for the computerization of factories, this is a revolution that will spread across industries globally. It is predicted that the adoption of Industry 4.0 will benefit production due to increased connectivity across entire industry as manual factories are transformed into smart factories. Just like steam power, electricity, and digital automation of the past, cyber-physical systems will create the factory of the future: the smart factory.

By using the cyber-physical system, it can help the factories with increased connectivity between management level and the production floor. This means people are able to monitor the manufacturing process in real time and make decentralized decisions based on data feedback through the network. The autonomous networking and systems along with the cloud platform help predict maintenance issues or system failures and react to them accordingly. Saving valuable time and money for companies. These technologies are also revolutionizing the way things are designed, demand for mass production and even product lifecycles.

The evolution of the Internet of Things has gone beyond internet connected appliances in recent years with the integration of different technologies such as machine learning, embedded systems, and wireless connection. The Internet of Things is pivotal to the inception and evolution of Industry 4.0. The sensors in networked physical connected devices collect data in real time that is valuable to industry and can help optimize manufacturing processes.

In our project, we aim to use the concept of Industry 4.0 to monitor the CNC machine. Under CNC Machining, machine tools function through numerical control. A computer program is customized for an object and the machines are programmed with CNC machining language (called G-code) that essentially controls all features like feed rate, coordination, location and speeds. With CNC machining, the computer can control exact positioning and velocity. CNC machining is used in manufacturing both metal and plastic parts. But the disadvantage of CNC machine is people can only monitor the machine physically. So, the purpose of our project is to solve this problem, this means people do not have to monitor the machine physically, the can monitor the machine in everywhere as long as there is a network connectable device. People can even check the condition of the machine by tracking back to the data that they have received from the sensor.
1.2 Objective of Project

The objective of the project is to equip a CNC machine with sensors that can detect the machine's operating condition and transmit the sensed data to a cloud for further processing, such as detecting the temperature, humidity, current and vibration. The detected data will be send to the cloud, so that people can view the data from any devices, for example PC, mobile phone, or any other internet-connected electronic devices. These series of operations are called Industry 4.0.

In our project, we aim to install the sensors to the CNC machine. After the sensors are installed, it can sense all the data and the changes when the machine is operating. All the sensed data will be uploaded to the cloud through the embedded controller or computer. Hence, people can just monitor the machine by looking at the data that had uploaded to the cloud from any devices.

![Figure 1. Picture of how the project works](image)
1.3 Project Scope

In order to let our project works, we need to choose the suitable sensors that can detect well and correctly of the performance of the CNC machine. We decided to use temperature sensor, vibration sensor, humidity sensor and current sensor. Installation to the CNC machine of these sensors will be part of project too. We have to connect all these sensors to the embedded controller/computer, so the embedded controller/computer is able to upload the datas that is collecting by the sensors to the cloud platform.

We are require to know the knowledge of the embedded controller/computer, and the way of uploading datas from embedded controller/computer to the cloud platform. Finding the suitable embedded controller/computer and cloud platform will also be part of our project. In the end of this project, our sensors must be able to sense the right datas, collect the datas by embedded controller/computer and successfully upload to the cloud platform.

People must be able to review the performance of the CNC machine by login to the cloud from any devices. This project is useful and more productive as it can help people to monitor the machine at anywhere, anytime. People do not have to go to a specific place just to see the progress of the machine. If there is anything wrong in the operation, people can also track the datas easily by login to the cloud.
Research of Components

2.1 Sensors

A sensor is an electronic device that can detect and respond to some types of input from physical environment. These inputs that have detected by the sensor could be the heat, light, pressure moisture or any one of a great number of the physical environment. The output of the sensors is usually some signals that are convertible to human-readable data display at the sensor location or transmitted electronically over a network for reading or further processing. In our project, we will be using temperature sensor, current sensor, humidity sensor and vibration sensor to detect the performance of the CNC machine. ([1]TechTarget, 2012)

**Temperature sensor** is used to measure the changes of the temperature of certain object or environment. It plays an important role in many application, in our project, temperature detection is used as part of preventative reliable. For example, while an appliance may not actually perform any high temperature activities, the system itself may be at risk to overheating. This risk arises from specific external factors such as a harsh operating environment or internal factors like self-heating of electronics. By detecting when overheating occurs, the system can take preventative action. In these use cases, the temperature detection circuit must be reliable over the expected operating temperature range for the application. ([2] Ametherm, n.d.)

The TMP36(shows in Figure 2) is a low voltage, precision centigrade temperature sensor. It provides a voltage output that is linearly proportional to the Celsius temperature. It also doesn’t require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range. It is very easy to use: Just give the device a ground and 2.7 to 5.5 VDC and read the voltage on the V out pin. The output voltage can be converted to temperature easily using the scale factor of 10 mV/°C. ([3] Spark fun Electronic, n.d.)

![Figure 2. Temperature sensor - TMP36](image-url)
The purpose of **humidity sensor** (shows in Figure 3) is to measure and report the relative humidity in the machine. Humidity is the presence of water in air. The amount of water vapour in air can affect many manufacturing processes in industries. The presence of water vapour also influences various physical, chemical, and biological processes. (4) Engineersgarage, n.d.)

![Figure 3. A humidity sensor](image)

The **current sensor** (shows in Figure 4) is used to measure the voltage that has been used by the machine. Current sensors are either open- or closed-loop. Open-loop current sensors measure AC and DC currents and provide electrical isolation between the circuit being measured and the output of the sensor. Open-loop current sensors are generally preferred in battery-powered circuits given their low-operating power requirements and small footprint features. Closed-loop sensors measure AC and DC currents and provide electrical isolation. They offer fast response, high linearity, and low temperature drift. The current output of the closed-loop sensor is relatively immune to electrical noise. The Closed-Loop sensor is sometimes called a ‘Zero-Flux’ sensor because its Hall-Effect sensor feeds back an opposing current into a secondary coil, wound on the magnetic core to zero the flux produced in the magnetic core by the primary current. Closed-loop sensors are often the sensor of choice when high accuracy is essential. (5) DigiKey, 2012)
By measuring the range of vibration, a **vibration sensor** (shows in Figure 5) is needed. The tasks of the vibration sensor is to monitor the vibration and also can maintenance based on sensor data to avoid future issues. Vibrations produced by industrial machinery are vital indicators of machinery health. Vibration analysis is used as a tool to determine a machine’s condition and the specific cause and location of problems, expediting repairs and minimizing costs. Machinery monitoring programs record a machine's vibration history. Monitoring vibration levels over time allows prediction of problems before serious damage can occur. ([6] DigiKey, 2012)
2.2 Embedded Controller/Computer

Arduino Uno (shown in Figure 6) is a microcontroller motherboard. A microcontroller is a simple computer that can run one program at a time, over and over again. It is best used for simple repetitive tasks such as opening and closing a garage door, reading the outside temperature and reporting it to Twitter, driving a simple robot. And is very easy to function too. ([7]RaspberryRights, 2015)

![Arduino](image)

Figure 6. Arduino

Raspberry Pi (shown in Figure 7) is a general-purpose computer, usually with a Linux operating system, and able to run multiple programs. It is best used when you need a full-fledged computer like driving a more complicated robot, performing multiple tasks, doing intense calculations (as for Bitcoin or encryption). But it is more complicated to use than an Arduino. ([8] RaspberryRights, 2015)

![Raspberry Pi](image)

Figure 7. Raspberry Pi
2.3 Types of Interface

In general, an interface is a device or a system that unrelated entities use to interact. For example, a remote control is the interface between human beings and television, languages is the interface between people. In our project, in order to connect the sensors and the embedded controller/computer, interfaces is needed.

The Inter-integrated Circuit (I²C) Protocol is a protocol intended to allow multiple “slave” digital integrated circuits (“chips”) to communicate with one or more “master” chips. Like the Serial Peripheral Interface (SPI), it is only intended for short distance communications within a single device. ([10] SparkFun, n.d.)

There is some terminology for I²C, the device that transmit data to the bus is called transmitter, the device that receives data from the bus is called receiver. Master is the device that generates clock, starts communication, sends I²C command and stop communication, slave is the device that listens to the bus and is addressed by the master. Multi-master means I²C can have more than one master and each can send commands. Arbitration is a process to determine which of the masters on the bus can use it when more masters need to use the bus. Synchronization is a process to synchronize clocks of two or more devices. ([1] i2cinfo, n.d.)

Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers. With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices. ([10] Arduino, n.d.)

There are some advantages about SPI, SPI is faster than asynchronous serial and the receive hardware can be a simple shift register, it also supports multiple slaves. ([11] SparkFun, n.d.)

However, it requires more signal lines (wires) than other communications methods. The communications must be well-defined in advance (you can’t send random amounts of data whenever you want), and the master must control all communications (slaves can’t talk directly to each other). It usually requires separate SS lines to each slave, which can be problematic if numerous slaves are needed. ([11] SparkFun, n.d.)
2.4 Cloud Platforms

It is an information technology paradigm, a model for enabling ubiquitous access to shared pools of configurable resources, which can be rapidly provisioned with minimal management effort, often over the Internet. In our project, we will do some research about the most common use cloud platforms and then decide the most suitable one.

2.4.1 ThingSpeak Cloud Platform

ThingSpeak is an Internet of Things (IoT) platform that lets you collect and store sensor data in the cloud and develop IoT applications. The ThingSpeak IoT platform provides apps that let you analyze and visualize your data in MATLAB, and then act on the data. Sensor data can be sent to ThingSpeak from Arduino, Raspberry PI, BeagleBone Black, and other hardware. ThingSpeak is also one of the most suitable cloud platforms for our project. ([16] Sdx Central, n.d.)

2.4.2 Cayenne Cloud Platform

Cayenne can customize your online and mobile dashboard with drag-and-drop widgets. Able to take full control of the project by creating a custom dashboard with drag-and-drop widgets to visualize, manage, and control connected devices. Add and remotely control sensors, actuators, and more to single board computers, microcontrollers, and other hardware connected to Cayenne. Widgets are used to visualize devices data, status, and actions. Every device, sensor and actuator that is added in Cayenne has one or more widgets associated depending on the hardware capabilities. ([18] myDevice, n.d.)
2.5 Selected Components

After a week of researched, we chose to pick Raspberry PI. We chose it as it is easier in terms of connecting the wiring between the Raspberry PI and the sensors, connecting the Raspberry PI to the cloud and also handling the programming. Raspberry PI can also run multiple programs.

![Raspberry Pi](image.png)

Generally, accelerometers contain capacitive plates internally. Some of these are fixed, while others are attached to miniscule springs that move internally as acceleration forces act upon the sensor. As these plates move in relation to each other, the capacitance between them changes. From these changes in capacitance, the acceleration can be determined. Accelerometers with a digital interface can either communicate over SPI or I2C communication protocols. These tend to have more functionality and be less susceptible to noise than analog accelerometers.([13] SparkFun, n.d.)

Next, is the selection of sensors, which we chose the temperature, the vibration and the current sensors. The temperature sensing that we have pick is DS18B20, the vibration sensing is ADXL345 and the current sensor is SCT013. The reason we picked temperature, vibration and current sensors because we trying to get the temperature in the CNC machine when the machine is working, we also planned to keep track of the coolant temperature when it is running. For the vibration sensors, which is called the accelerometers, we planned to keep track of the vibration motions while for the current sensors, is for us to know the current drawn from the CNC machine. As we already chose our sensors, from there we started to learn about the wiring between the Raspberry PI and the sensors.
Figure 9. Vibration Sensor

Figure 10. Temperature Sensor
The next selection will be the choice of choosing which cloud platform we wanted. Which we have picked Cayenne and Thingspeak. This is due to the easy and fastest excess between the controllers. By choosing these two cloud platforms, we get started with the python programmes for the Raspberry Pi to send the data to the cloud platforms.
Connections

3.1 Raspberry Pi

The function of the Raspberry PI in this project is work as an controller, so in order for it to work, we need to connect different types of sensors to it for collecting the data, and connect it to the monitor to do the setting and programming for the controller and sensors.

Pi-EzConnect (shows in Figure 12) makes it possible to connect multiple sensors to multiple GPIOs on a Raspberry PI, and it is very useful in our case, since we are connecting three sensors to the Raspberry PI. ([19] adafruit, n.d.)

Figure 12, Pi-EzConnector.
For temperature sensor, we are using the waterproof version of the DS18B20 so the device has three leads, red, black and yellow. The DS18S20 has three legs. Two of these pins put the sensor in a circuit between a high voltage pin and ground. This connection provides the power for the sensor to measure the temperature. The third pin connects to a GPIO pin, and it is with this connection that we can communicate with the sensor. Before connecting the sensor to the Raspberry PI, we are suppose to switch off the Raspberry PI first. If not, there will be electricity running through the pins. The DS18B20 sensor needs to be connected with the black wire to ground, the red wire to the 3V3 pin and the yellow wire to the GPIO4 pin. During the connection we need a 4.7k to 10k resistor, the resistor is used as a 'pullup' for the data-line, and is required to keep the data transfer stable, it needs to be connected between the 3V3 and GPIO4 pins. ([20] leanpub, n.d.) (Refers to Figure 14)
The Raspberry Pi has drivers for one wired devices to be connected to GPIO pin-4 by default. 1-wire is similar to I²C with longer range and low data rates. It can be used to communicate with inexpensive devices like thermometers, humidity sensors and other 1-wire sensors over a long range. 1-wire ranges approximately 150m while I²C ranges approximately 100m. The main advantage is since each device has a unique address, any number of devices can be connected on a single wire limited by the drive capacity of Raspberry Pi’s GPIO and the total capacitance seen on the line. The 1-wire bus requires a weak pull up resistor as all devices linked to this bus via a tri-state or open drain output. Here a 4.7KΩ resistor is used as pull up. The pull-up resistor assures that the wire is at a defined logic level even if no active devices are connected to it.
After the connection, we are supposed to turn the Raspberry PI on, then put our finger against the sensor. If it is connected the wrong way round, then it will get very hot within a second or two, in which case, turn the Raspberry PI off once it has booted, and wait a while for it to cool down, and then take out the sensor and put it back in again the right way round. We do need a PI-EzConnect saves us from having to attach wires directly to the Raspberry PI board, as this would just be really fiddly. It also has better labelling on the GPIO pins, which makes it easier to work with.

Next, is to read the temperature from the sensor. We are supposed to open the terminal and type python. At the python IDLE command prompt, type in all the commands. (Refer to Appendix A)

In the commands just now, the text is slowly broken down until just the temperature values remain, with several lines of code. This many lines of code is not necessary, splitting by spaces from the start, not newlines will also work. The code may be harder to read if it is squashed onto one line, however. This operation can be made faster.

After that we need to turn off the Raspberry Pi again. Obtain an LED, a 220Ω resistor, and also some more wires. Pick one of the numbered pins on the ribbon cable that connects your Pi to the breadboard and connect the long leg (on the opposite side to the flat edge of the LED base) of the LED to this pin. Connect the 220Ω resistor between the ground pin of the Pi and the other (longer) leg of the LED. The purpose for this is to set up a circuit from one of the GPIO pins, through the LED and resistor, down to ground. With the Pi, we can switch on a specific GPIO pin, so it has a high voltage, and current will flow through the LED and resistor. The resistor we
have put in stops too much current flowing, which could heat up the LED so much that it would burn out.

Then we turn on the raspberry pi again. Open a root terminal from the menu, or by running `sudo su` in a normal terminal. Now, type these commands, where "18" in this example is replaced by whichever pin the LED is connected to:

```bash
cd /sys/class/gpio/
echo "18" > export
cd gpio18
echo "out" > direction
echo "1" > value
```

With the LED, we wanted to change something about the hardware (make the GPIO pin we had chosen into an output), which could lead to bad things happening if we changed something we shouldn't have, or changed the wrong pin number. With the Raspberry Pi, it is designed in a way so that you can't break anything just by changing the wrong GPIO pin, but the restriction is left over from more expensive computers that are more easily broken. Once we've open a root terminal, first we have to tell the Pi that we want to do something with a particular GPIO pin, which we do by using `echo` to write the pin number into the export file. The Pi notices we have done this, and sets up a folder for that pin to allow us to control it. Then we tell the Pi that we want this pin to be an output pin, again using `echo`, so that we can turn the LED on and off. Writing the number "1" to the value file makes the pin turn on, so that current flows out of that pin. Once we have got our LED working, we turn it off by echoing "0", and then telling the Pi that we are done working with that pin by writing the pin number to the unexport file. (21) cl.cam, n.d.)
For vibration sensor (ADXL345) is a small, thin, ultra low power, 3-axis accelerometer with high resolution (13-bit) measurement up to ±16 g. The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-Fall sensing detects if the device is falling. These functions can be mapped to interrupt output pins. Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

Digital output data is formatted as 16-bit two’s complement and is accessible through either an SPI (3- or 4-wire) or I²C digital interface. There are four wires with the accelerometer: black, red, green and yellow. For the wiring part, the black wire will connect to the GND, the red wire will connect to the 3.3V, and the green and yellow wire will connect to G2/P3 and G2/P5 of the PI-EzConnector. (shows in Figure 18)
In order to use I2C, we are required to do some configuration on the raspberry pi, can refer to Appendix B for the detail of configuration. ([23] Anstack, n.d.)

The Raspberry PI has a great range of connection points and we will begin our set-up of the device by connecting quite a number of them to appropriate peripherals. The Raspberry PI needs to store the Operating System and working files on a micro SD card, so a SD card is needed. Figure 19 shows the position of putting a SD card. ([21], Leanpub)

A keyboard and mouse is needed too, as we need them to do programming after the sensors are connected. (shows in Figure 20)
The Raspberry PI also comes with the HDMI port, means we are able to connect it to the monitor with an HDMI connection. (shows in Figure 21)

The Raspberry PI have a standard RJ45 network connector on the board ready to go. In a domestic installation this is most likely easiest to connect into a home ADSL modem or router.(shows in Figure 22)
There is also a port for power supply, shown in Figure 23.

In Figure 24, shown an overall summary of what interface each sensor used for the operation of collecting the data.
Figure 24 Types of interface used
3.2 Connection between Raspberry PI and Cloud

We have selected two cloud platform to try out and connect to the Raspberry PI, which is Cayenne and ThingSpeak. By proving the Raspberry PI had successfully connected to the cloud, we need to get the data from those sensors that had already attached to the raspberry pi. Previously, we do connected the temperature sensor and the vibration sensor to the Raspberry PI. So, in this section we will show the details and failures that we had go through during connection.

3.2.1 ThingSpeak Cloud Platform

The first step is to sign up an account in ThingSpeak, create a channel for our data, so that we can monitor our data from the channel. After we done creating channel in ThingSpeak, we move onto our next step, which is connecting the raspberry pi to the cloud. By connecting the Raspberry PI to the cloud, we need to get the API key and the channel ID. We paste the API key and Channel ID to the python programme that we have created previously in the raspberry pi.

![API key and Channel ID](Image)

Initially, we did not manage to connect the ThingSpeak to the raspberry pi, as there were some errors in our programme. So, ends up we do get our data for the temperature sensor, but is it only display on the Raspberry PI computer. It is not uploaded to the cloud platform, shown in Figure 26.
Due to the python programming, if we miss up any little thing, it can cause an error. The first trying we failed is because our coding is not exactly correct. But eventually we manage to find out our mistake. On the second try, we realised we keyed in the wrong number for our API key. After we successfully got to see the data from the cloud platform. We move on to try out the accelerometer. It was much easier for us as we got the experience from connecting the temperature sensor of Raspberry PI to the ThingSpeak cloud platform.
We created a new channel with two fields, so that we can monitor all the data from both sensors at the same. For the python programming, we try to combine the coding for temperature sensor and the coding for the accelerometer together. So, in the end we manage to get both data from two sensors at the same time.

![Figure 28 Data for both temperature and vibration sensor](image)

3.2.2 Cayenne Cloud Platform

The first step we did for the connection is to sign up an account in the Cayenne. Next is downloading the Raspbian system package and copying it into the SD card. We then install the Raspberry Pi the Raspbian system. For the Raspbian installation, it is recommended to use a HDMI screen, a USB mouse and a USB keyboard. After this, we connect the Raspberry Pi with the LAN cable.
Then we open the Cayenne app and install the library on the device. We add a device on the Cayenne app and select the Raspberry Pi on the screen to pick out single board computer.
It took a few minutes to display as the Raspberry PI took some time with the installations and to load the data onto the device. It was successful as the data collected from the sensor has shown onto the device, shown in Figure 32.

We then tried to give a test on whether the temperature sensor is working properly by changing the temperature of the room. It took some time like the previous reading, but in the end the results were correct, shown Figure 33, the temperature decreases as the time goes by.
Figure 33 Chart of the updated temperature sensor
Installation

During the installations, we got help from our lecturer, Dr Win Tun Latt, and Mr Ramadass, a research engineer as we were being cautious and not wanting to damage any parts of the CNC machine. With this being said, we began by preparing the parts to mount on the CNC machine.

Figure 34, Measurement of the machine
In Figure 35, shows that the current sensor is put and lock onto the live wire, with the action done, it is able to measure the current drawn out for the CNC machine. This allows us to keep track of the power supply then has been drawn.

*Figure 35 Current sensor put onto the live wire*
In Figure 36, shown how the temperature sensor is mounted inside the CNC machine.

In Figure 37, is the temperature sensor for the coolant. The temperature sensor cable must be long enough to go around the back of the CNC machine. Hence, we have extended the wiring for the temperature sensor as it was too short before. With this, the cable is able to be connected with the Raspberry PI.
In Figure 38 and Figure 39, shown the assembled parts of the temperature sensor for the coolant. The objective of the sensor is to measure the temperature of the coolant before and after the machine have been operated.
As shown in Figure 40, it is the cable way to the Raspberry PI. After we extended the cable, the length is more sufficient and there is excess to it. We also keep the cable under the machine as we want to avoid any tripping of cables. If the cable is expose outside, it may lead to a trip and fall hazard.

Next is the vibration sensor, we made a bracket for the sensor so that it can be mount on the machine. We use metal base plate as the first option, so that it can be mounted and be secured. Hence, we drilled and tap the bracket and assembled them together shown in Figure 42 and Figure 42.
Figure 41 Tapping the bracket

Figure 42 Vibration Sensor on the bracket
After mounting the vibration sensor, we tried to take the reading of the vibrating sensor, but it got an error. We troubleshoot on the problem. We found out that it can’t be placed on a magnetic based plate, so we tried putting on some blu tack over the sensor to cover it up but still failed.

We chose to change the bracket to another material. We changed it to an alkalic bracket as it a good conductor of electricity. In Figure 44, is the modified bracket for the vibrating sensor. We used blu tacks and plastic glue around the bracket to seal and secure any gaps or holes.
Figure 44 Modified Vibration Sensor Bracket

Figure 45 Vibration and Temperature Sensors
After changing the base plate and mounting up the sensors, we test the sensors again to check if there is any errors. The data is shown below in Figure 48 and Figure 48.

<table>
<thead>
<tr>
<th></th>
<th>Coolent Temperature at Source</th>
<th>Coolent Temperature at Nozzle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>3:43:00 PM</td>
<td>23.10</td>
<td>24.90</td>
</tr>
<tr>
<td>3:43:15 PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:43:30 PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:43:45 PM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 47 Temperature Sensor Readings
After getting all the datas and done with the installation, we actually connected our cloud platform to the software called Power BI. Power BI desktop is a feature-rich data mashup and report authoring tool. Combine data from disparate databases, files, and web services with visual tools that helped to understand and fix data quality and formatting issues automatically.
Figure 50 Overview of Page
Problem Faced

The problem that we faced in the project is basically getting the codings right, to display the data that the cloud platform has collected. We repeat the process of redoing the coding a few times as it does not display the right data. In the end we managed to get the results that we wanted. Another problem we faced is choosing the right material to make the bracket for the vibration sensor as we do not want to damage the sensor itself. We chose the alkalic base plate as it is a good conductor of electricity and solved the problem.
Recommendation

The aim of our project is to sensorise the machine, and make it Industry 4.0. So, by doing this project we realised there is still room for improvement in our project. For example, we can install a camera into the machine. This means that people not only can determine the conditions of the machine based on the data they have received. They can also look at the live stream video through the Power BI. This video can also be recorded, so if there is anything happen and the engineer or technician wanna track back to find what happened, they can check by looking at the recorded video.

Figure 51. Live stream video of the machine
In order to have live stream video, we decided to use the wirecast live stream software. Wirecast is a live video streaming production tool. It can control real-time switching between multiple live video cameras. So, what we need to do is install a camera into the machine, and wirecast will connect to it, the wirecast can also connect to the Youtube. We only need to create a channel in Youtube, and make sure the wirecast is connected to Youtube channel. Then we insert the Youtube live video link into the Power BI. So, people are able to monitor the condition of the machine through Power BI. In the future, we can also install more cameras into the machine, so that we can monitor it from different angle.
Appendix A

# Open the file that we viewed earlier so that python can see what is in it.
Replace the serial number as before.
tfile = open("/sys/bus/w1/devices/10-000802824e58/w1_slave")
# Read all of the text in the file.
text = tfile.read()
# Close the file now that the text has been read.
tfile.close()
# Split the text with new lines (\n) and select the second line.
secondline = text.split("\n")[1]
# Split the line into words, referring to the spaces, and select the 10th
word (counting from 0).
temperaturedata = secondline.split(" ")[9]
# The first two characters are "t=", so get rid of those and convert the
temperature from a string to a number.
temperature = float(temperaturedata[2:])
# Put the decimal point in the right place and display it.
temperature = temperature / 1000
print temperature
Appendix B

From our Raspberry Pi, we need to install:

```
sudo apt-get install python-smbus i2c-tools
```

Enable the I2C kernel module in the Raspberry Pi:

```
sudo raspi-config
```

Edit the modules file (sudo vim /etc/modules) and make sure it contains the following lines:

```
i2c-bcm2708
i2c-dev
```

Remove I2C from the blacklist file (/etc/modprobe.d/raspi-blacklist.conf) commenting the following line if it appears:

```
#blacklist i2c-bcm2708
```

After all these previous steps, reboot the Raspberry Pi

```
sudo reboot
```

Test the connection to the I2C module:

```
sudo i2cdetect -y 1
```

The command should print the following output:

```
00: 01 02 03 04 05 06 07 08 09 a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15
```

When having the module working properly, we need to get and use the python ADXL345 library to have access to the time based data.

Clone the library repository and execute the example code:

```
git clone https://github.com/pimoroni/adxl345-python
cd adxl345-python
sudo python example.py
```

The command's output should be:

```
ADXL345 on address 0x53:
x = 0.0240
y = -0.3046
z = 0.8246
```
## Appendix C

<table>
<thead>
<tr>
<th>Description of activities</th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
<th>Week 5</th>
<th>Week 6</th>
<th>Week 7</th>
<th>Week 8</th>
<th>Week 9</th>
<th>Week 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary discussion on proposed project topics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brainstorm Research</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information gathering (per interface, sensors, cloud platforms)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare for FYP Preliminary Presentation (Report &amp; PowerPoint slides)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FYP Preliminary Presentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Selections</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assemble of Raspberry Pi (Connect to sensors, controller &amp; devices)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workshop 1 (Measuring for installation at CNC Lab)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description of activities</th>
<th>Week 11</th>
<th>Week 12</th>
<th>Week 13</th>
<th>Week 14</th>
<th>Week 15</th>
<th>Week 16</th>
<th>Week 17</th>
<th>Week 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workshop 2 (Measuring for installation at CNC Lab)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workshop 3 (Presentation at CNC Lab)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Editing/Testing of Raspberry Pi (With Cayenne Cloud Platform)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Editing/Testing of Raspberry Pi (With Thingspeak Cloud Platform)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References

1. Available from: http://whatis.techtarget.com/definition/sensor
   (Accessed July 2012)


   (Accessed 13 Sept 2012)

   (Accessed 18 Oct 2012)

   (Accessed 4 Dec 2015)

   (Accessed 4 Dec 2015)


10. Available from: https://learn.sparkfun.com/tutorials/i2c

11. Available from: http://i2c.info/i2c-bus-specification


14. Available from:
   https://learn.sparkfun.com/tutorials/accelerometer-basics

15. Available from:
   https://learn.sparkfun.com/tutorials/analog-to-digital-conversion

16. Available from:
    (Accessed June 2015)

17. Available from:
    https://www.sdxcentral.com/projects/thingspeak/

18. Available from:

19. Available from:
    https://mydevices.com/cayenne/features/

20. Available from:

21. Available from:
    https://leanpub.com/RPiMRE/read

22. Available from:
    https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/temperature/

23. Available from:
    https://www.anstack.com/blog/2016/07/05/accelerometer-intro.html.
Plagiarism Declaration

I am aware of the SP penalty for plagiarism, which is that if I am guilty of plagiarism in one module I will fail all my modules in the semester, or even be liable for expulsion. I understand that plagiarism means passing off another person’s work as my own, without crediting the person.

I/we declare that the work I am/my team is submitting is my/our own. I/we have acknowledged all sources of information referenced.

Module & Module Code: ME3009
Assessment/project report: Industry 4.0: Sensorising CNC Machine
Submission Date: February 19, 2018

<table>
<thead>
<tr>
<th>Student Name/s:</th>
<th>Student ID/s:</th>
<th>Signature/s:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhao Zedong</td>
<td>P 1534436</td>
<td></td>
</tr>
<tr>
<td>Muhammad Aidil Bin Azman</td>
<td>P 1549962</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>

This form is to be attached to the assignment/project report to be submitted.