eISSN:2600-7920
INTI JOURNAL
Vol.1, 2018 (26)

A prototype of Storm Water Channel IoT (SWCI) to Prevent Flood Damage in Malaysia

Deshinta Arrova Dewi[1], Jonathan Wijaya[2], Kannan Ponkoodalingam[3]

[1,2,3] Faculty of Information Technology and Science (FITS), INTI International University, Nilai, Negeri Sembilan, Malaysia.

Corresponding Author: deshinta.ad@newinti.edu.my

Abstract

Malaysia as a nation of Southeast Asia has an average of raindrop 250 centimeters or 98 inches and an average of temperature 27°C or 80.6°F. Malaysia is naturally exposed to weather wonder affect like is El Nino that decrease the amount of rainfall during the dry season. It influences Malaysia by expanding the sea and ocean levels and rainfall, which prompted a huge area of droughts and endangered Malaysia with flooding. This leads Malaysia to do a pre-emptive action to prevent flood by building a SMART Tunnel (Stormwater Management and Road) in Kuala Lumpur. With the help of the SMART Tunnel, the prevention actions against flash flood are performed. However, a few upgrades are needed to increase the efficiency of the SMART tunnel upon traffic and resources. For example, the SMART tunnel will be closed if heavy rain occurs and approximately 1.000.000 m^3 of stormwater will flow to Kerayong River and then go to Klang River. Both rivers actually have been polluted due to heavy development in the city. Hence, there is a need to find solutions whereby the SMART tunnel will continuously open during heavy rain. This helps the traffic flow on the road. A proper mechanism is also needed to utilize the excess water and preserve it as water resources for daily life. This paper attempts to overcome the drawback of the current SMART tunnel in Kuala Lumpur by introducing a prototype of Storm Water Channel IoT (SWCI). The SWCI is designed to manage the stormwater channel to ease the traffic condition during raining or dry season. SWCI is connected to several reservoirs or a special tank so that the water can be preserved for daily purposes instead of channeling it to the rivers. The prototype of SWCI is developed using an Arduino microcontroller, pushbullet API, Blynk interface designer and Thingspeak cloud services.

Keywords

Storm Water Channel, Internet of Things, Arduino, SMART Malaysia

Introduction

According to the Science Magazine (Cohen, 2003), the urban population will be growing from 75 % to 83 % in developed countries and 40% to 56% in developing countries. This indicates the impervious surfaces will also be expanded with the environmental impacts. Consequently, with increased rainfall expected, the drainage infrastructure in cities will become an exceptional demand.

In this context, Best Management Practices (BMP) as alternatives of Low Impact Development (LID) may mitigate the problem and ensure greater longevity of drainage systems. Besides, improving the quality of water that reaches urban streams and rivers, as quoted in the study in Portland, USA (City of Portland, 2009). According to bio-retention principles, infiltration and evapotranspiration techniques (DeBusk, 2011), BMP makes use of various typologies, recommend in stormwater management manuals (eg.: City of Portland, 2009).

A previous research has presented five principles of stormwater management for protecting the ecosystem (Walsh, C. J. 2016). These principles are broadly applicable to all urban landscape that drain to a receiving the stream. They are: 1) ecosystems to be protected and a target ecological state should be explicitly identified; 2) post development balance of evapotranspiration, streamflow, and infiltration should mimic the predevelopment balance, which typically requires keeping significant runoff volume from reaching the stream; 3) the stormwater control measures (SCMs) should deliver flow regimes that mimic the predevelopment regime in quality and quantity; 4) the SCMs should have capacity to store rain events for all storms that would not have produced widespread surface runoff in a pre-development state, thereby avoiding increased frequency of disturbance to biota; and 5) the SCMs should be applied to all impervious surfaces in the catchment of the target stream.

Currently, SMART (Storm Water Management and Road Tunnel) is available in Kuala Lumpur. The main objective is to solve the problem of frequent flash flood and reduce traffic jams along Sungai Besi road and Loke Yew flyover at Pudu during rush hour. The SMART tunnel is the longest multi-purpose tunnel in the world that consists of two components: the *stormwater tunnel* and *motorway tunnel*. The SMART tunnels work with 4 modes of activations as depicted in Figure 1. First, the activation during normal condition which has no rain so the channel will operate as a normal traffic channel and there will be no water in the stormwater channel. Second, during a medium level of the storm, whereby the water will be diverted into the stormwater channel and it still works as a normal traffic channel. Third, traffic will be diverted because all the channel is going to be used for the stormwater channel. Fourth, the channel is used as the stormwater container which then flowed to Kerayong River.

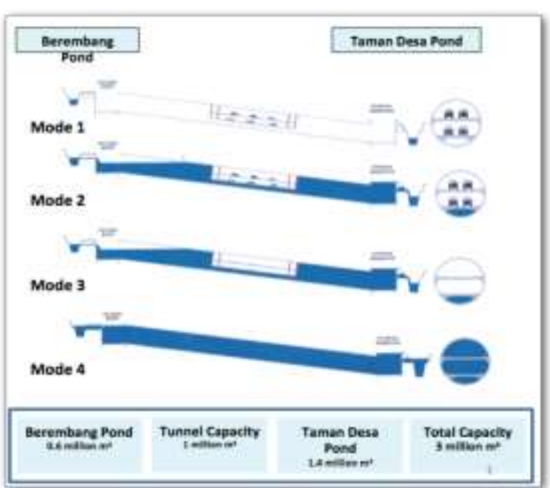


Figure 1. Smart Tunnel Mode (smarttunnel.com.my)

Although the current SMART tunnel has fully advantageous, there are few upgrades that can be done to increase the efficiency of the SMART tunnel upon traffic and resources. The following problems are stated in the SMART tunnel official website:

1. If the rain is heavy then the channel needs to be closed until the rain stopped and it needs 2 to 8 hours in mode 3 to reopen the channel which can disrupt the traffic.
2. The amount of water which is 1 million m^3 that is being stored while raining will flow to Kerayong River and then continue to Klang River which is polluted because of heavy development in the city.
3. The current stormwater channel doesn't have a reservoir to store the stormwater that can be used for resources. They only have a reservoir to store the water temporarily.

This paper attempts to overcome these drawbacks by proposing an implementation and a new upgrade to manage traffic and facilitates the excess water. The comparison between the current SMART tunnel and the proposed SWCI is presented in Table 1 below.

Current SMART Tunnel	**Proposed SWCI**
Prevent flood	Prevent flood
The tunnel will be closed if heavy rain occurs	Traffic will go as usual despite heavy rain
1.000.000 m^3 of stormwater will flow to Sg. Kerayong which will then flow to Sg.Klang which is reportedly polluted because of heavy development in the city	Stormwater will be saved as usable water resources which can be used for daily purposes.
No electronic gates for the water to flow to the bank which the current one didn't have any bank.	Electronic gates that connect to the water bank in the channel below the road that will be opened when there is rain so that the water can be stored.
No sensors were installed for the Software Monitoring System	Sensors will be installed which will work with the Electronic Gates and Software Monitoring System
CCTV Monitoring System	Software Monitoring System

Table 1. The comparison of the current SMART tunnel and SWCI

SWCI is designed with the help of the Internet of Things (IoT) due to lower cost in the development. The IoT sensors have offered a lower price than brand new networking solutions and powered battery like LORA. The battery-powered itself is considered as a generous saving resource compared with the costly electricity.

The objectives of this paper are listed as followings:

1. To study the strength and weakness of the SMART tunnel in Kuala Lumpur.
2. To develop SWCI that address the current issues in the SMART tunnel using the Internet of Things (IoT) technology.
3. To conduct testing of the proposed CSWI.

Methodology

The methods applied to carry out this project is outlined as below.

1. Requirement Gathering and Quick Design.

Literature reviews are the main activity at this stage to understand the strength and weakness of the SMART tunnel in Kuala Lumpur. Afterward, a review of IoT sensors and technology is performed to identify the accurate tools and configuration. Following this, the data collection and analysis is conducted using research instruments like questionnaires, document review, observation and interview with the relevant parties involved.

2. Building and Refining Prototype of CWSI.

In this stage, the installation of all relevant sensors and devices are performed that consists of TowerPro SG90 Servo, Water Sensor, DC3V-5V Submersible Miniature Water Pump Motor Ultra Quiet, and Water Proof DS18b20 Temperature Sensor Probe. All those devices need to be connected as modules that constitute SWCI device. The following figure 1 describes the module connection: a) Water sensor connection b) Servo motor connection, c) Temperature sensor module connection and d) Arduino Windshield for Ethernet connection.

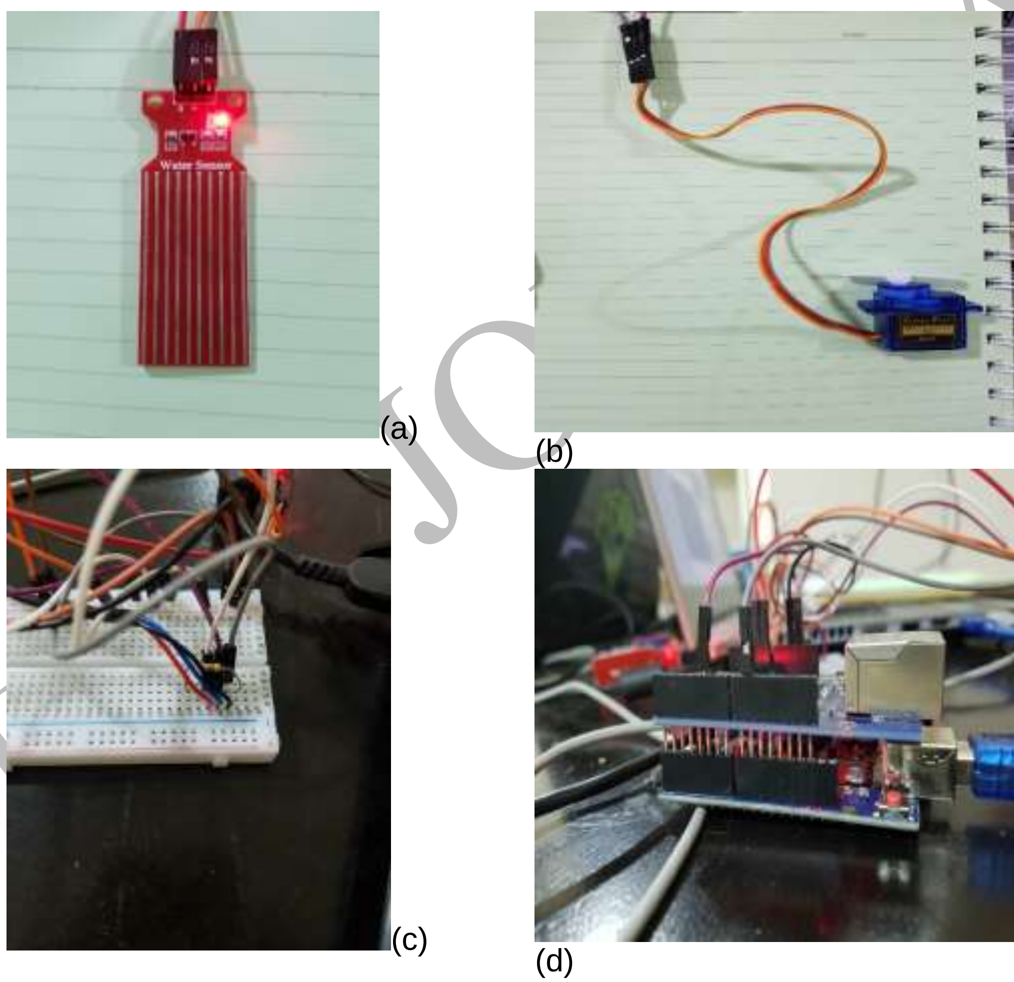
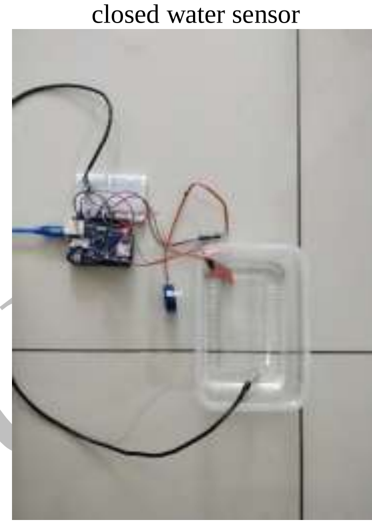


Figure 1. Device Tools and Connection for SWCI

The purpose of this water sensor is to detect water in the environment near the water sensor. The water sensor works if there are water detected in the metal piece planted in the water sensor. Hence, a water sensor program needs to be written using the codes extract as follows:

```
const int WaterSense = A0; //set water
sensor to A0
{
  int sensorValue =
analogRead(waterSens);//read the water
sensor value
  String x = String(sensorValue);


     if (sensorValue >= 100)
  {
    myservo.write(0);      // tell
servo to go to position in variable
'pos'
    delay(15);             // waits
15ms before the next code
  }


  else if( sensorValue <=100 )
  {
    myservo.write(180);    // tell
servo to go to position in variable
'pos'
    delay(15);             // waits
15ms for the servo to reach the
position
  }

  Serial.print("Requesting Water
Sensor Value: ");
  Serial.println("DONE");
  Serial.print("Sensor value is ");
  Serial.println(sensorValue);

  delay(5000);

       return x;
}
```

closed water sensor



Open water sensor

Figure 2. The programming code for the water sensor

Next is to identify the water level calibration using own formula as a result of the experiments. The experiments show the raw analog value of the water sensor is around 600 or 1200. In some cases when the silver plating is fully covered by water, it reaches a value of 600 and the length of the water sensor is 4cm. This finding shows that if the max value is 600 hundred it means that the water level is at 4cm. Hence, the formula for this experiment is:

$$4cm = \frac{600}{\text{x}} \tag{1}$$

This point value 4cm is not known by the Arduino programming. To find a digit of "x" for the program to automatically help the author find the water level. It means that the formula will be changed to this

$$x = \frac{600}{4\text{cm}} \tag{2}$$

It means that the value "x" is 150. The author can use the formula above which is

$$water\ level = \frac{\text{value from water sensor}}{150} \tag{3}$$

eISSN:2600-7920
INTI JOURNAL
Vol.1, 2018 (26)

Hence, the code extracts the can be used for water calibration is depicted in figure 3 below.

```
float waterLevel = outputValue/150 ;
const int waterSensorForPump = A2;

String waterLevelSensor()
{
  int              outputValue          =
analogRead(bankWaterLevelSensor);
  float waterLevel = outputValue/150 ;
  String z = String(waterLevel);
  outputValue   =   map(outputValue,   0,
1023, 0, 180);

  Serial.print("Requesting  Water  level
from water sensor: ");
  Serial.println("DONE");
  Serial.print("Water level is ");
  Serial.print(waterLevel);
  Serial.print(" cm");
  Serial.print("\n");

  delay(5000);

    return z;
}
```

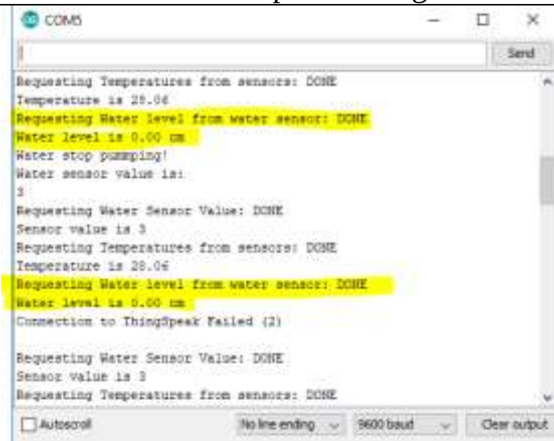


Figure 3. The water calibration code

The following figure 4 illustrate the overall prototype of SWCI after integration with Thingspeak, Pushbullet, and Blynk. With this integration, SWCI is able to send a notification to android mobile phone whenever water level is alarming.

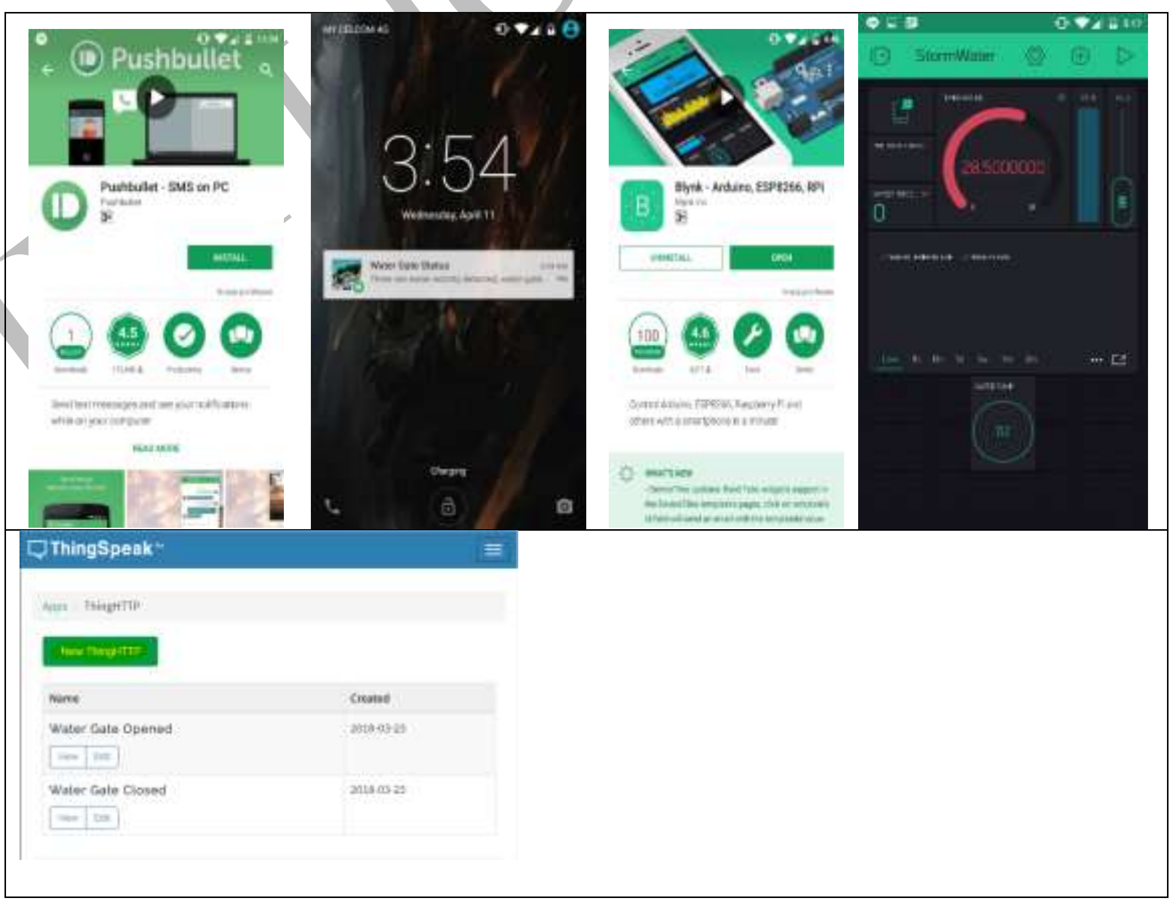


Figure 4. The final integration of SWCI

Results and Discussion

This section presents the results of the prototype development of SWCI. The testing comprises of water level sensor testing, water temperature sensor testing, and a water sensor to trigger the water pump. Another module name water pump Arduino is also tested, together with a water sensor in the second bank to stop water from pumping in the first water bank.

A water sensor consists of a series of traces connected to the ground. These traces are known as Ground Traces. Interlaced between the ground traces, there are traces known as Sensor Traces. The sensor traces are connected to the 1Mohm resistor. When water drops and shorts the sensor trace and the grounded trace, the resistor will pull the value of the sensor trace and identify it as HIGH. In this research, the water sensor has three main functions as stated in figure 5. Not only the water sensor detects water value in the first bank, but it also generates data and connects with third-party of cloud services i.e. ThingSpeak and Blynk.

Unit Name: Water Level Sensor	
Component	Water Sensor, Arduino
Function	1) To detect water value in the first water bank 2) To convert the value to the output desired which is 4cm max by using author's own formula for the calibration. 3) To upload the water level into ThingSpeak and author's interfaces which are Blynk.
Result	Successful.

Figure 5. The result of the water level sensor

Similarly, the water temperature is tested to know its functionality in detecting temperature in the first bank. The result is captured and displayed in Thingspeak and Blynk.

Unit Name: Water Temperature Sensor	
Component	DS18B20, Arduino
Function	1) To detect the temperature of the first water bank 2) To upload the temperature data into ThingSpeak and Blynk
Result	Successful.

Figure 6. The result of Water Temperature Sensor

Another module that handles a water sensor that is meant to activate the water pump is tested in figure 7. The excess water is channeled to the second bank in this process. This particular water sensor is the implementation of the proposed model that is proposed current SMART tunnel.

Unit Name: Water Sensor To Trigger Water Pump	
Component	Water Sensor, Arduino
Function	1) To detect water in the first water bank 2) To trigger the water pump to work if there are water detected from the water sensor 3) To upload water pump activity by using Binary language which is "1" is on when water is detected while "0" while water is not detected to ThingSpeak.

Result	Successful.

Figure 7. The result of the water pump and water sensor

As result of figure 7, the water pump itself is tested in figure 8 to ensure it receives the value from the water sensor and work accordingly. In the second bank, a water sensor is installed to stop water pumping. This module is important for energy save as it works only on certain condition, otherwise, it is set to be off. Figure 9 explain the result of the water sensor which placed in the second water bank.

Unit Name: Water Pump	
Component	Water Pump, Arduino
Function	1) To work accordingly to the water sensor 2) Or to work manually using the Blynk interface 3) To pump water from the first bank to second water bank
Result	Successful

Figure 8. The result of the water pump

Unit Name: Water Sensor in 2nd Bank to Stop Water from pumping	
Component	Water Sensor, Arduino
Function	1)To detect water in the 2nd water bank 2)If water is detected to a certain value then the water pump will stop pumping the water from first water bank to second water bank
Result	Successful

Figure 9. Water Sensor 2nd Bank

Conclusions

This project took approximately 10 months to complete and able to achieve all objectives. The project is divided into two sections: research and development. During the development stage, the technical problem like as Intel VT-X setting has been resolved. It requires efforts to troubleshoot and in some cases to find an alternative platform than Linux and Openhab. After further deliberation, the alternative tools i.e. ThingSpeak and Blynk were proposed and the development was able to continue till completed. The testing shows that SWCI is able to demonstrate the idea of the improved SMART tunnel problem in Kuala Lumpur. This project is adopted in a final year project for Bachelor in Computer Science (BCSI) at INTI International University.

References

Walsh, C. J., Booth, D. B., Burns, M. J., Fletcher, T. D., Hale, R. L., Hoang, L. N., & Wallace, A. (2016). Principles for urban stormwater management to protect stream ecosystems. *Freshwater Science*, *35*(1), 398-411.

Moura, N. C. B., Pellegrino, P. R. M., & Martins, J. R. S. (2016). Best management practices as an alternative for flood and urban stormwater control in a changing climate. *Journal of Flood Risk Management*, *9*(3), 243-254.

COHEN, Joel E. "Human population: the next half century". In: Science Magazine, Vol. 302. p. 1172- 1175.

CITY OF PORTLAND, Stormwater Management Manual. Revision 4, August 1, 2008. [http://www.portlandonline.com/bes/index.cfm?c=47952].

SMART Tunnel Operational Modes (2017). Retrieved from http://smartchannel.com.my/operational-modes/

Malaysia Faces Long-Term Water Crisis. (2017). Retrieved from https://www.wateronline.com/doc/malaysia-faces-long-term-water-crisis-0001

Sg Klang turns orange in color. (2017). Retrieved from https://www.nst.com.my/actionline/2017/08/273147/sg-klang-turns-orange-colour

Operational Modes. (2017). Retrieved from http://smarttunnel.com.my/operational-modes/

Malaysia | Facts, Geography, History, & Points of Interest. (2017). Retrieved from https://www.britannica.com/place/Malaysia

Merchant, C. (2017). Retrieved from http://www.expatgo.com/my/2011/09/08/weather-climate-and-geography-in-malaysia/.

Bellias, M. (2017). IoT for water utilities: intelligent water management. Retrieved from https://www.ibm.com/blogs/internet-of-things/iot-for-water-utilities