SASTA OLIPHANT SCIENCE AWARDS

SOUTH AUSTRALIAN SCIENCE TEACHERS ASSOCIATION

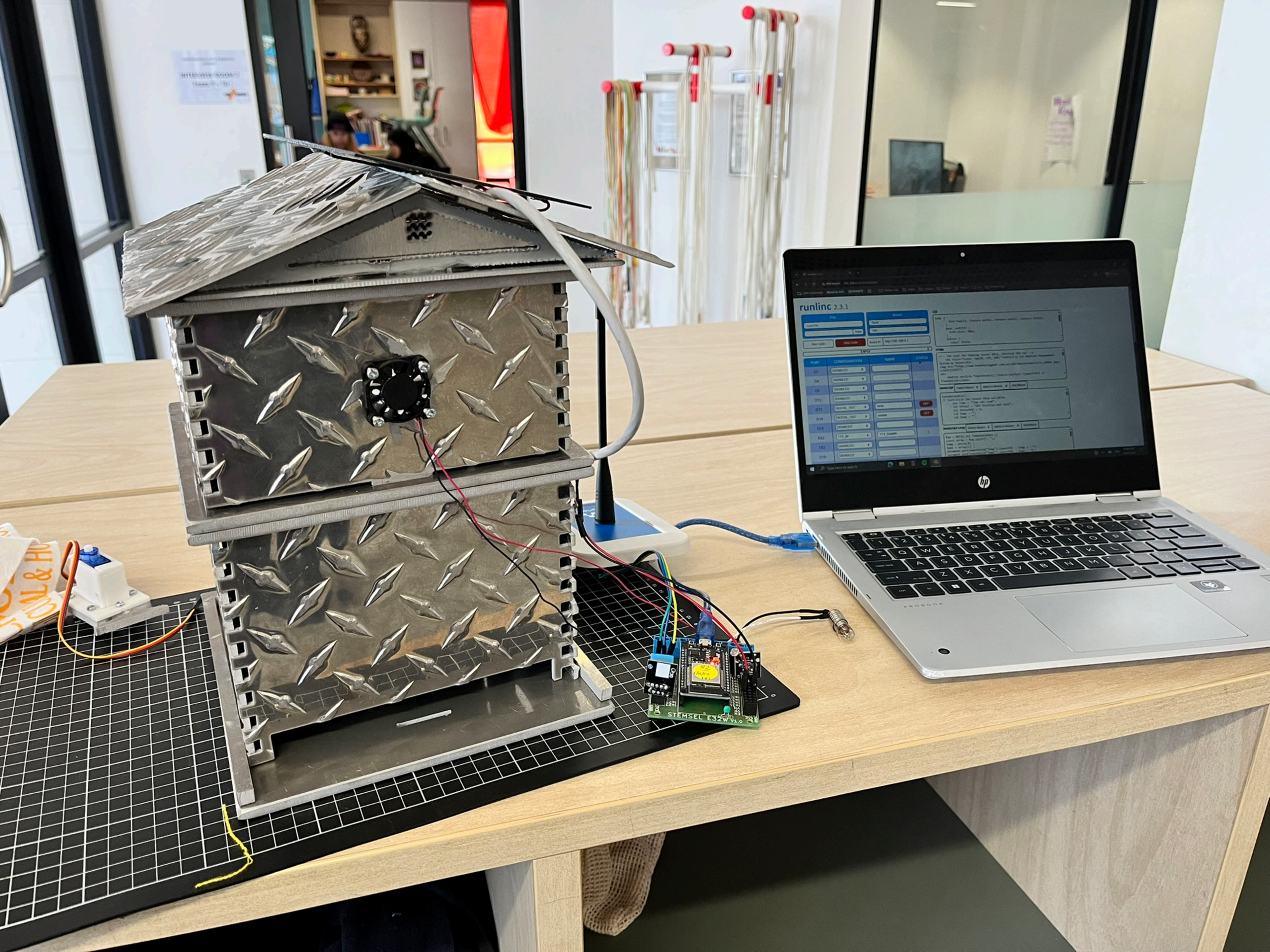
Highly Commended

Programming, Apps & Robotics

Year 9-10

Valar Rajesh

Glenunga International High School


runlinc 2.3.1
STEMSEL E32W V1.0

Beehive Bushfire Early Warning System

Introduction

The purpose of the Beehive Bushfire Early Warning System is for Bee Farmers who live in remote areas in Australia, who have to travel more than 4 hours every week to check on their beehives. In 2020, more than 10,000 hives were destroyed due to Bushfires according to the Australian Broadcasting Corporation. This system allows farmers to effectively save time, money, and effort by monitoring their beehives through their own webpage.

About the System

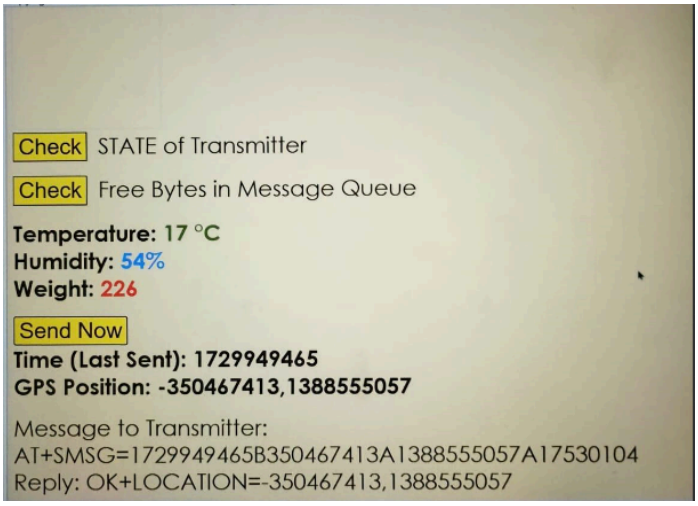
The Beehive Bushfire Early Warning System uses the runlinc controller and the myriota transceiver to receive and transmit data via a satellite, and then back to the farmers laptop. The system monitors three key features in the beehive, the temperature, humidity, and weight. These components are perfect for monitoring the health of the bees but also possible threats to the hive, such as bushfires. The controller collects the data from the sensors, and sends the readings to the transceiver. The transceiver then sends a signal to the satellite, which sends the data back to the farmers laptop. The farmer will be able to view all the data on their own webpage ,which will show the exact readings collected by the runlinc controller. This system has many advantages compared to others, as it is able to work in areas with no mobile network connection or internet; hence the reason why it is meant for farmers in remote areas.

Runlinc Wifi Controller: The controller is used to manage the sensors and collect data (temperature, humidity, and weight)

Myriota Transceiver: Transmits the data to back to the farmer's device vis the satellite

Sensors: Temp and humidity sensor, photoresistor(used to imitate the weight sensor)

Laptop or Computer: To access the webpage to view all the data



How to use the system:

1. Power the runlinc controller and connect it to the myriota transceiver.
2. Open the runlinc webpage on a browser-make sure the device is connected to the controller.
3. Click the "Run Code" button in the top left corner.
4. Scroll down to the bottom to view the webpage.

5. Press the “Check” button to see the state of the transmitter.
6. Press the “Send Now” button to transmit the latest data.

Hard Copy of the System

CSS:

```
body {
       font-family: Century Gothic, Century Gothic, Century Gothic;
     }
     body :not(h1) {
       font-size: 30px;
     }
     button {
       color: black;
       background-color: rgb(255, 255, 51);
       margin-right: 10px;
     }
img {
height : 200px;
width : 200px
}
body {
  background-color: #EEE8AA;
}

button
{ height:
75px; width:
175px;
```

HTML:

```
<!-- The code for showing senser data, checking the ses -->
   <h1 style="color: rgb(0, 119, 200)">Satellite IoT Beehive Management System by Valar</h1>
<img src="https://www.beekeepinggear.com.au/cdn/shop/products/11_2000x.jpg?v=1651377655">
<br>
   <button onclick="CheckState();">Check</button> <span>STATE of Transmitter</span>
   <br /><br />
   <button onclick="CheckQueue();">Check</button> <span>Free Bytes in Message Queue</span>
   <br /><br />

   <b>Temperature: <font id="Temp" color="green" ;></font></b>
```

```
  <br />
  <b>Humidity: <font id="Humi" color="blue"; ></font></b>
  <br />
  <b>Weight: <font id="SensorDisplay" color="red" ;> </font></b>
  <br /><br />

  <button onclick="Transmit();" id="sendButton">Send Now</button>
  <br />
  <b>Time (Last Sent): <font id="TimeReading"> </font></b>
  <br />
  <b>GPS Position: <font id="GPSReading"> </font></b>
  <br /><br />

  <span>Message to Transmitter: <font id="Message"></font></span>
  <br />
  <span>Reply: <font id="replyText"></font></span>

<br>

<h2><font id=RadNow></font> volts on solar panel</h2>

<br>

<h1>My Control Web Page
<button onclick="turnOn(fan);"><b>Fan ON</b></button>
<button onclick="turnOff(fan);"><b>Fan OFF</b></button>
<button onclick="turnOn(lamp);"><b>Lamp ON</b></button>
<button onclick="turnOff(lamp);"><b>Lamp OFF</b></button>
</h1>
<br>
```

Javascript:

```
turnOn(enable);
// Initialize the sensor data variables
    let Time = "Time not sent";
    let GPSval = "GPS Position not sent";
    let SensorVal = 0;
    let temp = "";
    let humi = "";

    async function Transmit() {
      GetTime();
      await mSec(5000);
```

```
  GetGPS();
  await mSec(5000);
  let message = generateFinalString(Time, GPSval, temp, humi, SensorVal);

  let ToSatellite = "AT+SMSG=" + message;
  document.getElementById("Message").innerHTML = ToSatellite;
  TTY_Out(ToSatellite + "\r");
}

function CheckState() {
  var ToSatellite = "AT+STATE=?";
  document.getElementById("Message").innerHTML = ToSatellite;
  TTY_Out(ToSatellite + "\r");
}

function CheckQueue() {
  var ToSatellite = "AT+MSGQ=?";
  document.getElementById("Message").innerHTML = ToSatellite;
  TTY_Out(ToSatellite + "\r");
}

function GetTime() {
  var ToSatellite = "AT+TIME=?";
  document.getElementById("Message").innerHTML = ToSatellite;
  TTY_Out(ToSatellite + "\r");
}

function GetGPS() {
  var ToSatellite = "AT+LOCATION=?";
  document.getElementById("Message").innerHTML = ToSatellite;
  TTY_Out(ToSatellite + "\r");
}

function generateFinalString(Time, GPSval, temp, humi, SensorVal) {
  SensorVal = SensorVal.toString();
  let latitude = GPSval.split(",")[0];
  let longitude = GPSval.split(",")[1];
  if (
    Time.length != 10 ||
    latitude.length > 10 ||
    latitude.length < 8 ||
    longitude.length > 11 ||
    longitude.length < 8 ||
```

```
  temp.length > 3 ||
  temp.length < 1 ||
  humi.length > 2 ||
  humi.length < 1 ||
  SensorVal.length > 3 ||
  SensorVal.length < 1
 ) {
  return "data length Invalid";
 } else {
  // Convert positive and negative numbers into the required data format
  // latitude
  if (latitude.startsWith("-")) {
   if (latitude.slice(1).length === 8) {
    latitude = "B" + "0" + latitude.slice(1);
   } else {
    latitude = "B" + latitude.slice(1);
   }
  } else {
   if (latitude.length === 8) {
    latitude = "A" + "0" + latitude;
   } else {
    latitude = "A" + latitude;
   }
  }
  // longitude
  if (longitude.startsWith("-")) {
   if (longitude.slice(1).length === 8) {
    longitude = "B" + "00" + longitude.slice(1);
   } else if (longitude.slice(1).length === 9) {
    longitude = "B" + "0" + longitude.slice(1);
   } else {
    longitude = "B" + longitude.slice(1);
   }
  } else {
   if (longitude.length === 8) {
    longitude = "A" + "00" + longitude;
   } else if (longitude.length === 9) {
    longitude = "A" + "0" + longitude;
   } else {
    longitude = "A" + longitude;
   }
  }
  // temp
```

```
        if (temp.startsWith("-")) {
          if (temp.slice(1).length === 1) {
            temp = "B" + "0" + temp.slice(1);
          } else {
            temp = "B" + temp.slice(1);
          }
        } else {
          if (temp.length === 1) {
            temp = "A" + "0" + temp;
          } else {
            temp = "A" + temp;
          }
        }
        // humi
        if (humi.length === 1) {
          humi = "0" + humi;
        }
        // SensorVal
        if (SensorVal.length === 1) {
          SensorVal = "000" + SensorVal;
        } else if (SensorVal.length === 2) {
          SensorVal = "00" + SensorVal;
        } else {
          SensorVal = "0" + SensorVal;
        }
       }
       return Time + latitude + longitude + temp + humi + SensorVal;
     }

function lampON(){
document.getElementById("img").src="http://runlinc.com/online/picLampON.jpg";
}
function lampOFF() {
document.getElementById("img").src="http://runlinc.com/online/picLampOFF.jpg";
}

function fanON(){
document.getElementById("img").src="http://runlinc.com/online/picLampON.jpg";
}
function fanOFF() {
document.getElementById("img").src="http://runlinc.com/online/picLampOFF.jpg";
}
```

Javascript Loop:

```
Raw = DHT11_In( TempHumidData );
const array = Raw.split(", ");
temp = array[0] ;
humi = array[1] ;
document.getElementById("Temp").innerHTML = temp + ' °C';
document.getElementById("Humi").innerHTML = humi + '%';

SensorVal = analogIn( LightSensor );
document.getElementById("SensorDisplay").innerHTML = SensorVal;


let TTYraw = TTY_In( TTY_DataIN );
if( TTYraw != "n" ){
  DataFromSat = TTYraw.slice(0,TTYraw.length-4);
  //Time = DataFromSat.slice(0,7);
  }
if( DataFromSat.slice(0,7) == "OK+TIME" ){
  Time = DataFromSat.slice(8);
  }
if( DataFromSat.slice(0,11) == "OK+LOCATION" ){
  GPSval = DataFromSat.slice(12);
  }

document.getElementById("TimeReading").innerHTML = Time;
document.getElementById("GPSReading").innerHTML = GPSval;

document.getElementById("replyText").innerHTML = DataFromSat;

await mSec( 1000 );

solar_voltage = (analogIn(RAD_Sensor)/255)*3.3;
document.getElementById('RadNow').innerHTML = solar_voltage.toFixed(2);
```

Explanation of the Key Sections:

CSS: The CSS code designs the style of the webpage, including the colours, fonts, and button sizes.

HTML: The HTML section builds the structure of the webpage, containing the buttons, labels and data from each of the sensors.

Javascript: The Javascript section handles the sensor data collection, allows it to communicate with the Myriota Transceiver and does the satellite transmission.

Javascript Loop: The Javascript loop code constantly updates the live sensor data, the GPS data, and displays the replies from the satellite.

Additional Support:
For this project, there was also external support that helped create this system. Firstly, the runlinc team provided the platform, runlinc device and extra learning resources that helped with the code. In addition, the support from Myriota was also very helpful as it provided technical support when using the Myriota Transceiver.

Conclusion:
In conclusion, the Beehive Bushfire Early Warning System is an efficient and innovative project that supports bee farmers and their hives. By using the Runlinc controller and Myriota Transceiver, this system monitors the temperature, humidity, and weight of the hive without needing mobile network or internet access. That way, if there is a bushfire for example, then the farmer will be warned prior and will be able to get there earlier and help their bees. Not only does this reduce the time spent every week having to check the hives, but also the effort and money. Therefore, this system shows how technology can be used in even agricultural scenarios to solve real-world problems.

Bibliography:
Australia's 'littlest livestock' to get helping hand as bees suffer following drought and fire 2020a, www.abc.net.au.

Australia's 'littlest livestock' to get helping hand as bees suffer following drought and fire 2020b, www.abc.net.au.

Myriota 2019, Myriota.

RAPID DEVELOPMENT platform for IoT, AI and STEM inventions n.d., runlinc.com.