arm

Lesson 3

Air quality sensor

arm School Program

Success criteria

- Understand what a **forever** loop does
- Understand what the air quality sensor measures
- Understand that if code is only to be executed if a condition is true, then an ‘if then else’ statement can be used
- Use a logical operator in a program
- Consider what other applications the sensors could be used for in a product

Sensors

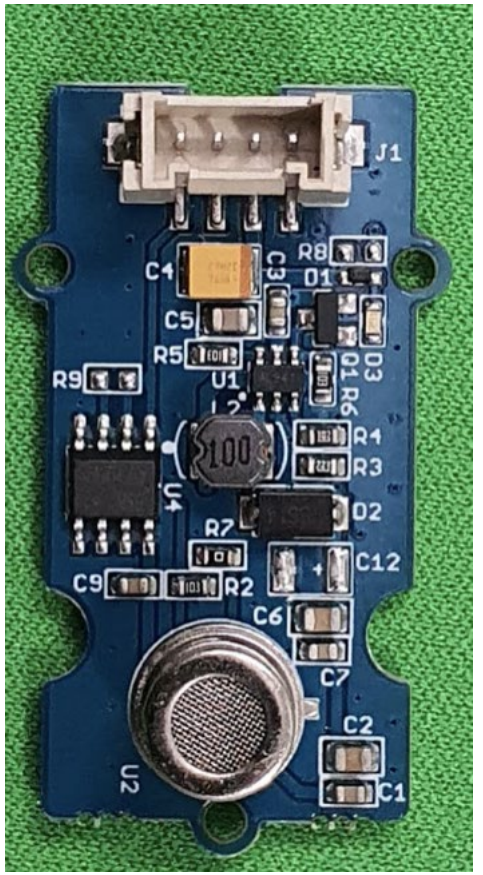
- **Sensors** allow a computer to detect certain measurements
- The measurements from a sensor will be processed and the micro:bit program can make decisions based on the value
- Sensors can be **analogue** or **digital**:
  - Digital sensors can only detect a single status; for example, if a temperature is above 50°C or not – they display the result as either a 1 or a 0
  - Analogue sensors continually measure values within a range

Types of sensor

- Different sensors can detect different things
- You are going to develop an air quality sensor which will upload readings to the Arduino cloud
- There are a range of different sensors which can detect a wide variety of different things
- Complete the Sensing your world worksheet, using the internet to help you

Air quality sensor

- The air quality sensor can detect harmful gases such as carbon monoxide, alcohol, acetone, thinners or formaldehyde



Program flow: sequence

- When code is run line by line, in the same order every time this is known as **sequence**

```
int quality = sensor.slope();
int reading;
ArduinoCloud.update();
reading=(sensor.getValue());
Serial.print("Sensor value: ");
Serial.println(sensor.getValue());
```

Program flow: selection

- Do computer programs always take the same route through the code?
- When code is only executed if a condition is true this is known as **selection**
- Complete the If, then, else worksheet

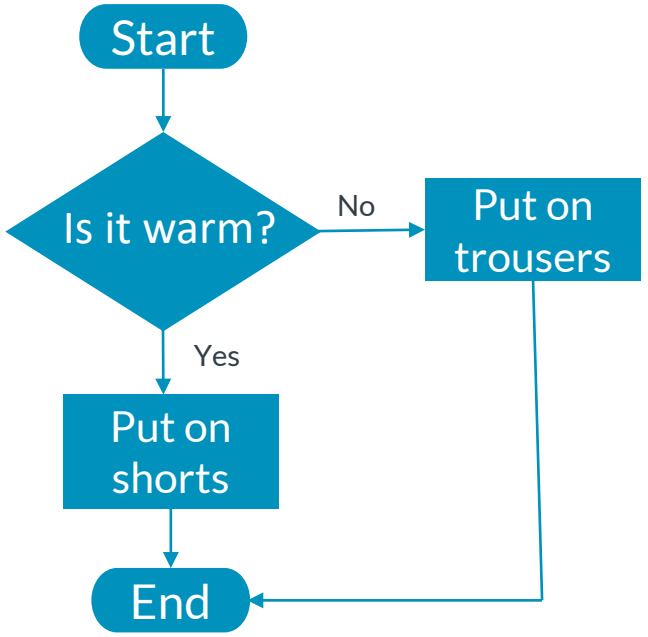
```
if (quality == AirQualitySensor::FORCE_SIGNAL) {
    Serial.println("High pollution! Force signal active.");
} else if (quality == AirQualitySensor::HIGH_POLLUTION) {
    Serial.println("High pollution!");
} else if (quality == AirQualitySensor::LOW_POLLUTION) {
    Serial.println("Low pollution!");
} else if (quality == AirQualitySensor::FRESH_AIR) {
    Serial.println("Fresh air.");
}
```

Program flow

- There will be times when a computer program will only need to do a certain task if a condition is true
- For example, think about getting dressed – what could cause you to dress differently?

Getting dressed algorithm

- Depending on the weather you may wish to dress differently. If we were to write this as an algorithm using a flow chart, what shape would we use for a decision?
- Our basic getting dressed algorithm could look like the flow chart opposite:
- As you can see from the flow chart:
  - You would only put on trousers if it is not warm
  - You would put on shorts if it is warm, taking a different route through the decision



Endless loops

- loop( ) is built into the Arduino language. It is the only one which loops forever
- If you program an Arduino to detect the Air Quality, it is important that it does this continuously. To do this, we use a loop.
- In the sample code, you can see that there is a delay of 1 second. This means that a value will be taken once per second. This can be adjusted as required.

```
void loop() {
    ArduinoCloud.update();
    int quality = sensor.slope();
    reading=int(sensor.getValue());
    Serial.print("Sensor value: ");
    Serial.println(sensor.getValue());
    Serial.print(reading);
    if (quality == AirQualitySensor::FORCE_SIGNAL) {
        Serial.println("High pollution! Force signal active.");
    } else if (quality == AirQualitySensor::HIGH_POLLUTION) {
        Serial.println("High pollution!");
    } else if (quality == AirQualitySensor::LOW_POLLUTION) {
        Serial.println("Low pollution!");
    } else if (quality == AirQualitySensor::FRESH_AIR) {
        Serial.println("Fresh air.");
    }

    delay(1000);
}


void onReadingChange() {
  // Do something
}
```

void loop() explained

- The void loop() runs all of the time and is a built in function
- The code measures a range of gases and outputs one of four different status:
  - High pollution! Force signal active.
  - High pollution!
  - Low pollution!
  - Fresh air
- int quality = sensor.slope() will calculate the slope of the line taken by the sensor and store it in the variable ‘quality’. This is then used in the if statement as the condition to determine which line should be executed
- reading=int(sensor.getValue() will take the measurement from the sensor and store in the variable ‘reading’
- The Serial.print(“Sensor value:”) will output the text string into the console
- The Serial.print(reading) will output the value stored in the variable ‘reading’ into the console

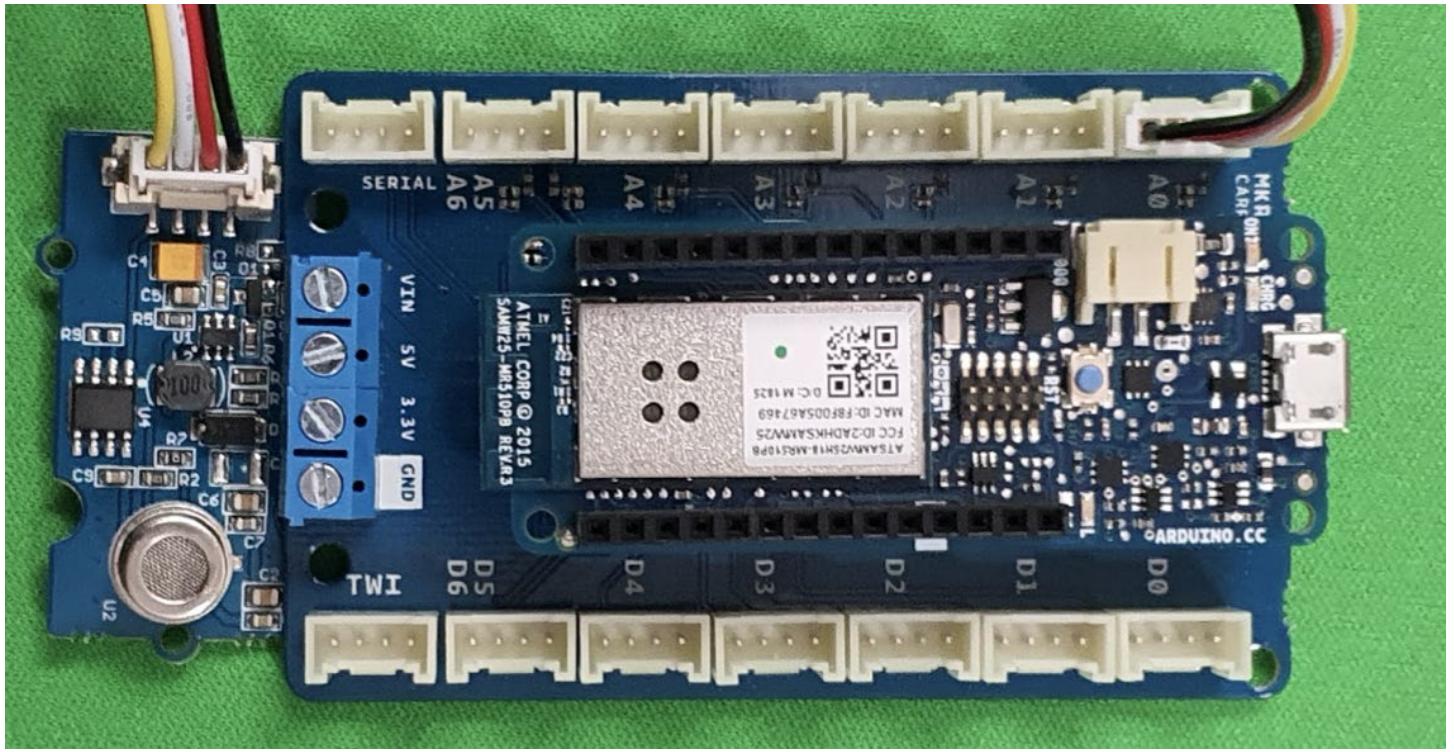
```
void loop() {
    ArduinoCloud.update();
    int quality = sensor.slope();
    reading=int(sensor.getValue());
    Serial.print("Sensor value: ");
    Serial.println(sensor.getValue());
    Serial.print(reading);
    if (quality == AirQualitySensor::FORCE_SIGNAL) {
        Serial.println("High pollution! Force signal active.");
    } else if (quality == AirQualitySensor::HIGH_POLLUTION) {
        Serial.println("High pollution!");
    } else if (quality == AirQualitySensor::LOW_POLLUTION) {
        Serial.println("Low pollution!");
    } else if (quality == AirQualitySensor::FRESH_AIR) {
        Serial.println("Fresh air.");
    }

    delay(1000);
}


void onReadingChange() {
  // Do something
}
```

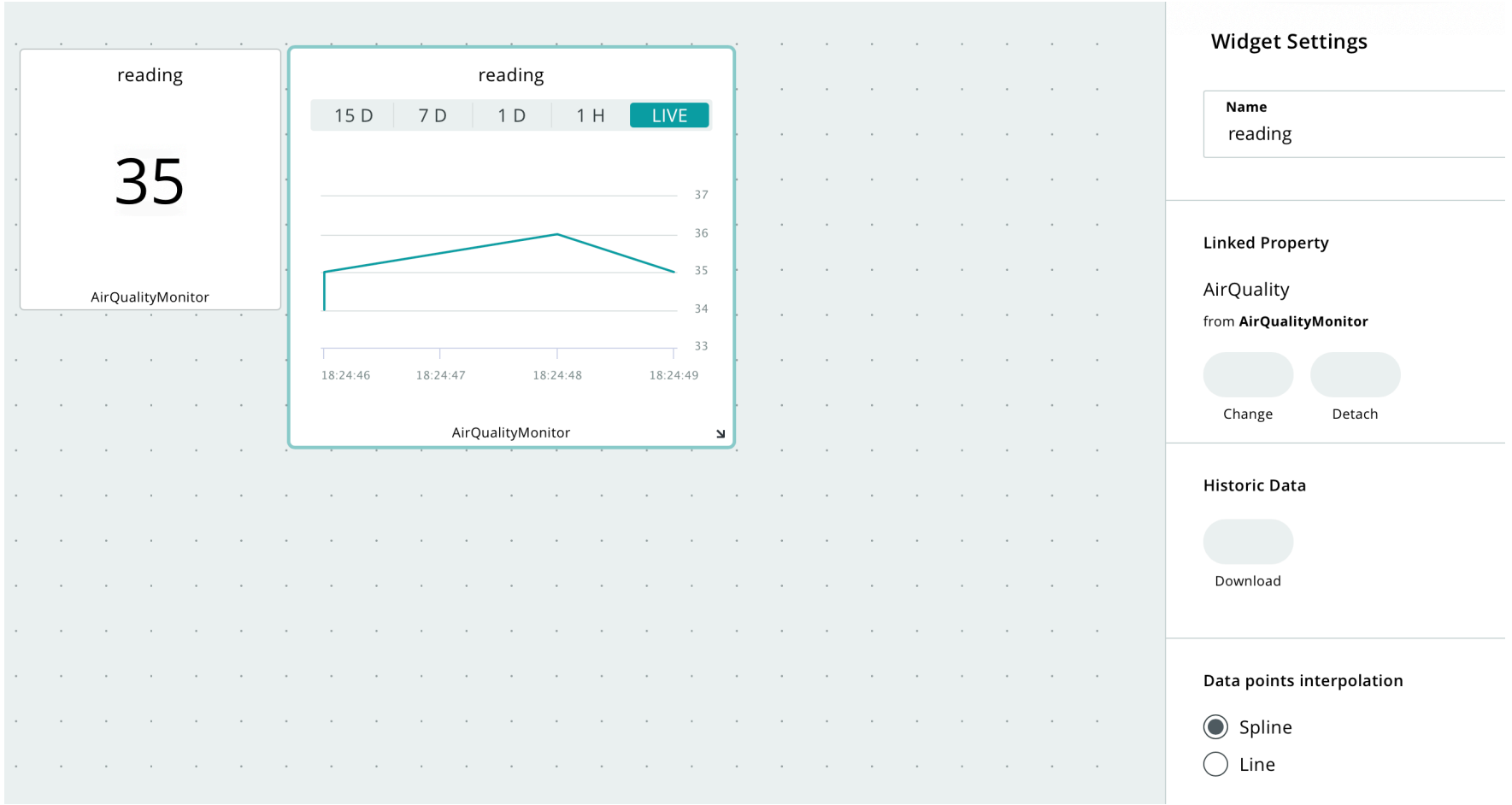
Assembling your device

- You should now follow the initial instructions on the worksheet to connect your device
- Be very careful not to bend any of the pins



Writing your code

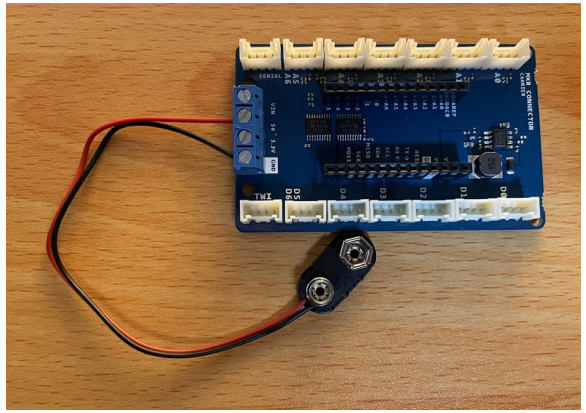
- Continue working through the worksheet to create your own air quality monitor



arm School Program

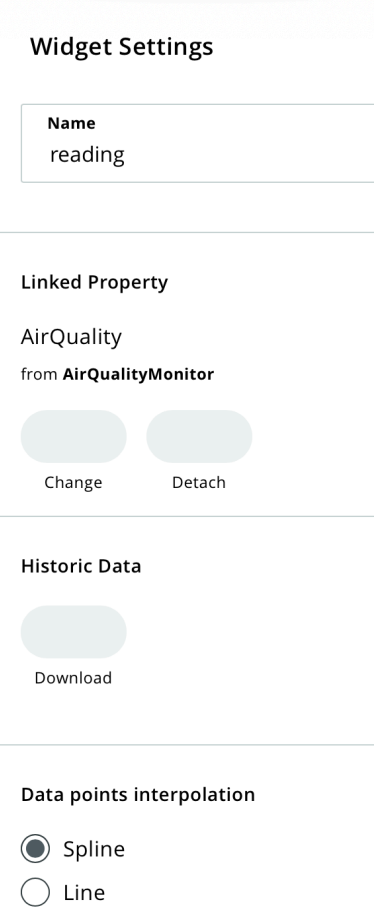
Extension activities

- You could create a waterproof case to hold your device so that you can monitor the air quality around your environment at different times of the day. Please note that the device will need to be within wifi range. You could do this using an old plastic margarine tub. You will need to cut out a hole roughly the size of the sensor so that readings can be taken. Mount the holder to that the hole where the sensor is mounted points downwards so that rain will not get in
- The device will also require a power source. You can use the battery connected provided with the kit. Connect the red wire to 5V and the black wire to GND.



Exporting the data as a CSV file for analysis

- A CSV file is a type of text file which separates the different values using a comma
- To download the readings into a CSV file click on ‘historic data’ and then ‘download’ from the IoT dashboard screen



Summary

- The Arduino IoT cloud can take readings from a range of sensors and display the data live on a dashboard
- If a branch of code is only executed if a condition is true, this is known as selection
- Loops within code is know as iteration

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
תודה

arm School Program