SASTA OLIPHANT SCIENCE AWARDS

SOUTH AUSTRALIAN SCIENCE TEACHERS ASSOCIATION

Prize Winner

Programming, Apps & Robotics

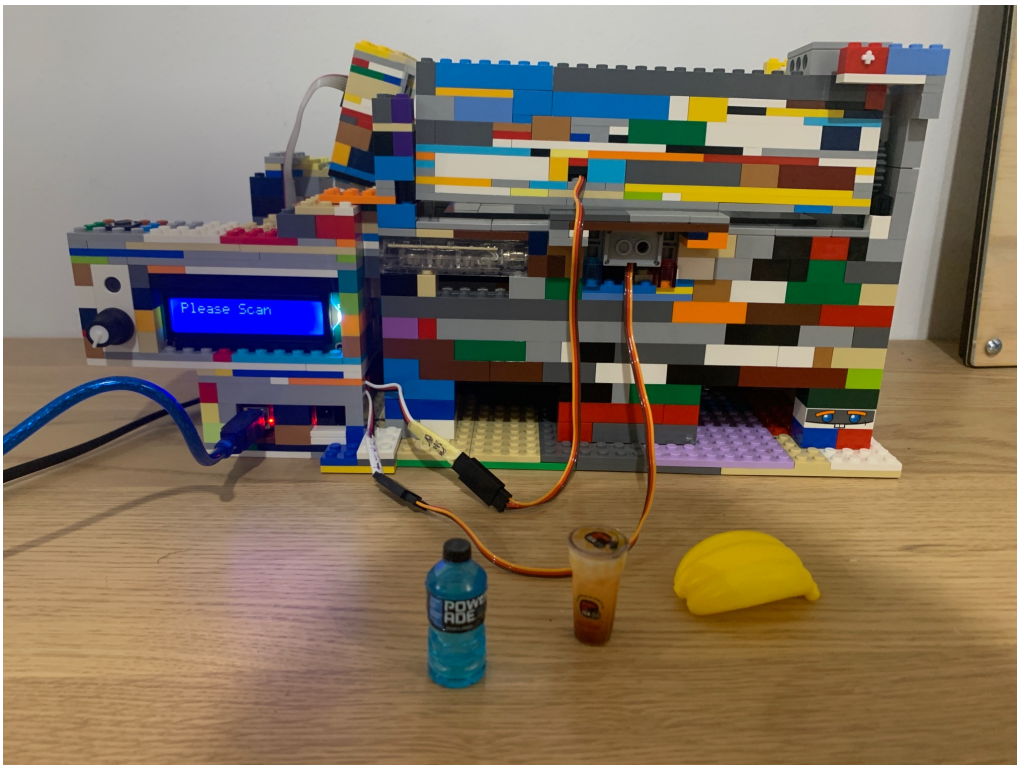
Year 3-4

Oaki Bellison

Belair Primary School

Rubbish Sorter 2.0

By Oaki Bellison



Introduction

It is important that people put their rubbish in the correct bin. People often don't put their rubbish in the right bin because they are not sure which bin the rubbish goes in. Two years ago I invented a rubbish sorter. You scanned the barcode on the rubbish and the correct bin it had to go in (landfill, recycling and soft plastic) would open.

I thought my original design needed an upgrade because some rubbish doesn't have a barcode like food, paper and coffee cups. It also required people reading the display which gave them instructions on which bin each item should go in. People still could have put rubbish in the wrong bin using my original rubbish sorter, because when more than one bin lid opened they might not be able to read or they may be someone who can't see properly or someone who doesn't want to follow instructions. Therefore I needed to make a new rubbish sorter to do the sorting automatically.

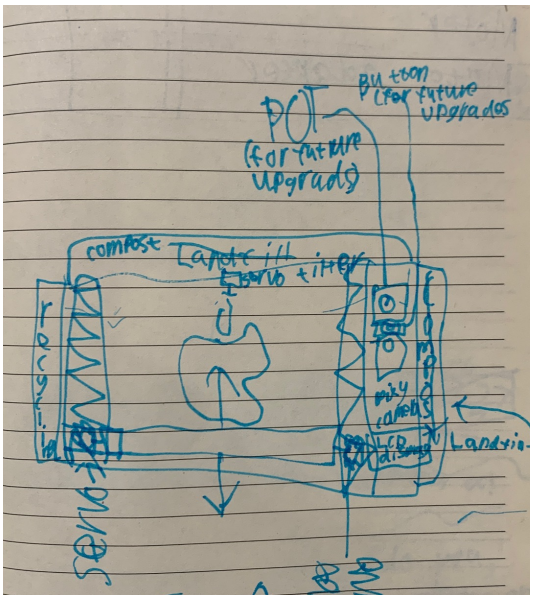
Most parks only have one bin: landfill. That means all the rubbish from the park goes to landfill. If compost goes to landfill, it can create greenhouse gases when it breaks down. If recyclables go to landfill, they wont be recycled, so more trees will be chopped down and more plastics will be made, which is bad for the planet.

Aim

To create a rubbish sorter that identifies people's rubbish using a camera and sorts it into the right bin, removing the possibility of human error in the rubbish sorting process.

Scientific Purpose

The scientific purpose of this project is to control a robot. This robot uses a colour sensor to get information about what the item of rubbish is, then responds to the information by tilting or pushing it into the correct bin.



An initial design sketch

Potential Applications

The Rubbish Sorter 2.0 could be used in parks, schools, workplaces, public spaces and homes. Having rubbish sorted by a robot, means humans can't make mistakes and put the rubbish in the wrong bin.

Type

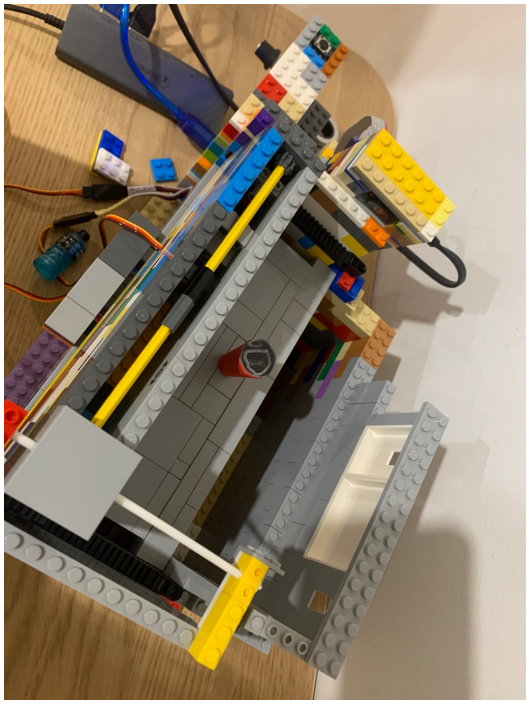
I created the Rubbish Sorter 2.0 with Lego and used an Arduino and Pixy2 Camera to scan and sort rubbish. I programmed it with ArduBlock and Arduino IDE. The Pixy2 Camera couldn't be coded using ArduBlock and I'm still learning how to code with Arduino IDE so I had to code in ArduBlock then add the code into Arduino IDE. I had to get help from some people from Robotics Club with writing the code for the Pixy2 Camera.

Instructions

Simply put the rubbish on the platform and the Rubbish Sorter 2.0 will sort it for you. It will tilt one way to sort into the landfill bin. It will tilt the other way to sort into the recycling bin. For the compost bin, there is a “pusher” that pushes the rubbish into the compost bin. On the screen, it tells you what rubbish you have put on the platform and what bin it goes in, so it can teach you the correct bin to put your rubbish in for non rubbish-sorting bins. The Rubbish Sorter just needs to be plugged into a power supply.

On the screen it tells you what rubbish you have put on the platform and what bin it goes in, so it can teach you the correct bin to put your rubbish in for non rubbish-sorting bins.

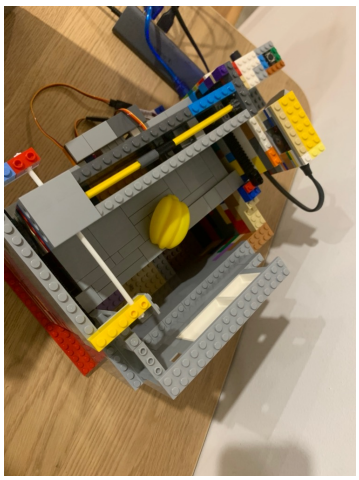
The Rubbish Sorter just needs to be plugged into a power supply.



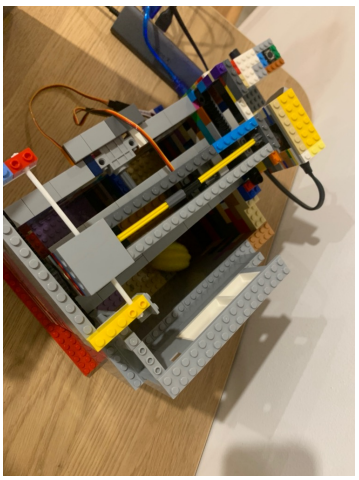
A coffee cup is placed on the platform

The platform tilts and the coffee cup falls into the landfill bin

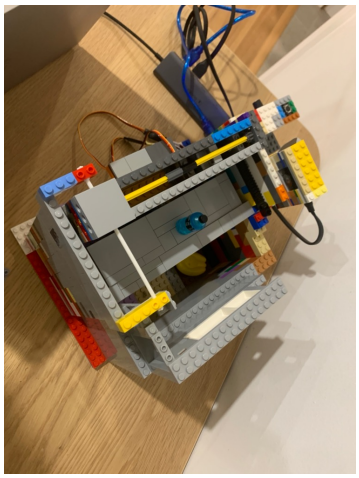
Instructions



A banana peel is put on the platform



The pusher pushes it into the compost bin



A plastic bottle is placed on the platform

The platform tilts and the bottle falls into the recycling bin

The Program

```
#include <LiquidCrystal.h>


#include <SPI.h>

#include <Pixy2.h>


// This is the main Pixy object

Pixy2 pixy;

#include <Wire.h>

#include <LCD.h>

#include <LiquidCrystal_I2C.h>


LiquidCrystal_I2C lcd_I2C_26(0x26, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);



int _ABVAR_1_ScannedRubbish = 0 ;

int _ABVAR_2_Tilter = 0 ;

int _ABVAR_3_Pusher = 0 ;

int _ABVAR_4_a;

int _ABVAR_5_BinActive = 0 ;

void __ardublockDigitalWrite(int pinNumber, boolean status)

{

  pinMode(pinNumber, OUTPUT);

  digitalWrite(pinNumber, status);

}


int _ABVAR_5_apple = 0 ;

int _ABVAR_6_plastic = 0 ;

int _ABVAR_7_A4paper = 0 ;

int _ABVAR_8_A4paper = 0 ;

int _ABVAR_9_A4cardboard = 0 ;

int _ABVAR_10_A4cardboard = 0 ;

int _ABVAR_11_recplastic = 0 ;

LiquidCrystal_I2C lcd_I2C_27(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

int _ABVAR_12_0 = 0 ;

int _ABVAR_13_a;

int _ABVAR_14_a;
```

This part of the program gets a few things ready for the rest of the code

```
void Compost();
void Landfill();
void Scan();
void Recycling();
void whatrubbish();
void Home();

void setup()
{
 Serial.begin(115200);
 Serial.println("Started!");
 pixy.init();
 lcd_I2C_26.begin (16, 2);
  lcd_I2C_26.setBacklight(HIGH);
  lcd_I2C_27.begin (16, 2);
  lcd_I2C_27.setBacklight(HIGH);
  lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
  lcd_I2C_26.print( "Loading..." );

  _ABVAR_1_ScannedRubbish = 0 ;

  _ABVAR_2_Tilter = 2 ;

  _ABVAR_3_Pusher = 3 ;

  _ABVAR_5_BinActive = 0 ;

  lcd_I2C_26.clear();

  lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
  lcd_I2C_26.print( "Welcome to the" );

  lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
  lcd_I2C_26.print( "Rubbish Sorter!" );

  delay( 2000 );

  lcd_I2C_26.clear();
```

This part of the program does the same thing and also makes the display say “Welcome to the Rubbish Sorter!”

```
  lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
  lcd_I2C_26.print( "Model: BHIDRS2" );


  lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
  lcd_I2C_26.print( "OS: BHID RS2 2.0" );


  delay( 1500 );


  lcd_I2C_26.clear();


  Home();
  CompostR();
}


void loop()
{
  ReadPIXY();
  whatrubbish();
}


void whatrubbish()
{
  {
    if (( ( _ABVAR_1_ScannedRubbish ) == ( 2 ) ))
    {
      lcd_I2C_26.clear();
      Compost();
      lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
      lcd_I2C_26.print( "Banana Peel =" );
      lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
      lcd_I2C_26.print( "Compost" );
      delay( 2000 );
      lcd_I2C_26.clear();
      Home();
      CompostR();
    }
    else
    {
      if (( ( _ABVAR_1_ScannedRubbish ) == ( 3 ) ))
```

This acts like a shortcut to putting the servos in their positions

This is like the shortcut to moving the servos but instead it links to a different part of the code that reads what the Pixy 2 camera can see and then identifies which rubbish it is.

This is what happens when the shortcut ”Whatrubbish();” is found in the code. It happens right after it reads what the Pixy2 camera can see and tells the servos which bin to put the rubbish in. It is called a subroutine.

This puts it in the compost

The LCD display teachers people the correct bin

This returns the servos to their home position

```
if (( ( _ABVAR_1_ScannedRubbish ) == ( 3 ) ))
{
  lcd_I2C_26.clear();
  Compost();
  lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
  lcd_I2C_26.print( "Bio Coffee Cup =" );
  lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
  lcd_I2C_26.print( "Compost" );
  delay( 2000 );
  lcd_I2C_26.clear();
  Home();
  CompostR();
}
else
{
  if (( ( _ABVAR_1_ScannedRubbish ) == ( 4 ) ))
  {
    lcd_I2C_26.clear();
    Recycling();
    lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
    lcd_I2C_26.print( "Bottle =" );
    lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
    lcd_I2C_26.print( "Recycling" );
    delay( 2000 );
    lcd_I2C_26.clear();
    Home();
    CompostR();
  }
  else
  {
    if (( ( _ABVAR_1_ScannedRubbish ) == ( 5 ) ))
    {
      lcd_I2C_26.clear();
      Recycling();
      lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
      lcd_I2C_26.print( "Can =" );
      lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
      lcd_I2C_26.print( "10 Cent Recycling" );
      delay( 2000 );
```

```
  lcd_I2C_26.clear();
  Home();
  CompostR();
}
else
{
  if (( ( _ABVAR_1_ScannedRubbish ) == ( 6 ) ))
  {
    lcd_I2C_26.clear();
    Landfill();
    lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
    lcd_I2C_26.print( "Coffee Cup =" );
    lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
    lcd_I2C_26.print( "Landfill" );
    delay( 2000 );
    lcd_I2C_26.clear();
    Home();
    CompostR();
  }
  else
  {
    if (( ( _ABVAR_1_ScannedRubbish ) == ( 7 ) ))
    {
      lcd_I2C_26.clear();
      Landfill();
      lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
      lcd_I2C_26.print( "Chip Packet =" );
      lcd_I2C_26.setCursor( (1) - 1, (2) - 1 );
      lcd_I2C_26.print( "Landfill" );
      delay( 2000 );
      lcd_I2C_26.clear();
      Home();
      CompostR();
    }
    else
    {
      if (( ( _ABVAR_1_ScannedRubbish ) == ( 0 ) ))
      {
```

```
                lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
                lcd_I2C_26.print( "Please Scan" );
                delay( 200 );
              }
              else
              {
                lcd_I2C_26.setCursor( (1) - 1, (1) - 1 );
                lcd_I2C_26.print( "Please Scan" );
                delay( 200 );
              }
            }
          }
        }
      }
    }
  }
}
```

End of the subroutine "Whatrubbish():"

This subroutine gets info from the Pixy2 Camera

```
void ReadPIXY()
{
// Replacement ReadPIXY subroutine;
 Serial.print("ReadPIXY Start ");
 _ABVAR_1_ScannedRubbish = 0 ;
  int i;
  // grab blocks!
  Serial.print("ReadPIXY GetBlocks Start ");
  pixy.ccc.getBlocks();
  Serial.print("ReadPIXY GetBlocks Done ");

  // If there are detect blocks, print them!
  if (pixy.ccc.numBlocks)
  {
    Serial.print("Detected ");
    Serial.println(pixy.ccc.numBlocks);
    for (i=0; i<pixy.ccc.numBlocks; i++)
    {
```

```
      Serial.print(i);
      Serial.print(": ");
      pixy.ccc.blocks[i].print();
      if (pixy.ccc.blocks[i].m_signature==1){
         _ABVAR_1_ScannedRubbish = 1 ;
      }
      if (pixy.ccc.blocks[i].m_signature==2){
         _ABVAR_1_ScannedRubbish = 2 ;
      }
      if (pixy.ccc.blocks[i].m_signature==3){
         _ABVAR_1_ScannedRubbish = 3 ;
      }
      if (pixy.ccc.blocks[i].m_signature==4){
         _ABVAR_1_ScannedRubbish = 4 ;
      }
      if (pixy.ccc.blocks[i].m_signature==5){
         _ABVAR_1_ScannedRubbish = 5 ;
      }
      if (pixy.ccc.blocks[i].m_signature==6){
         _ABVAR_1_ScannedRubbish = 6 ;
      }
      if (pixy.ccc.blocks[i].m_signature==7){
         _ABVAR_1_ScannedRubbish = 7 ;
      }
    }
  }
Serial.print("ReadPIXY End ");
// add more signatures above. max 7 sigs
}

void Landfill()
{
  for (_ABVAR_13_a=1; _ABVAR_13_a<= ( 20 ); ++_ABVAR_13_a )
  {
    __ardublockDigitalWrite(_ABVAR_2_Tilter, HIGH);
    delayMicroseconds( 2426 );
    __ardublockDigitalWrite(_ABVAR_2_Tilter, LOW);
    delay( 20 );
```

Each object has a signature that is from 1 to 7.
This sets a variable that other parts of the code use to know what rubbish is on the platform.

End fo “ReadPIXY();”

Subroutine for putting rubbish into the landfill bin.

```
  _ABVAR_5_BinActive = 2 ;
}

void Compost()
{
  for (_ABVAR_14_a=1; _ABVAR_14_a<= ( 100 ); ++_ABVAR_14_a ) // change (here)
  {
    __ardublockDigitalWrite(_ABVAR_3_Pusher, HIGH);
    delayMicroseconds( 2500 );
    __ardublockDigitalWrite(_ABVAR_3_Pusher, LOW);
    delay( 20 );
  }
  _ABVAR_5_BinActive = 3 ;
}

void Home()
{
  for (_ABVAR_13_a=1; _ABVAR_13_a<= ( 30 ); ++_ABVAR_13_a )
  {
    __ardublockDigitalWrite(_ABVAR_2_Tilter, HIGH);
    delayMicroseconds( 2080 );
    __ardublockDigitalWrite(_ABVAR_2_Tilter, LOW);
    delay( 20 );
  }
  _ABVAR_5_BinActive = 2 ;
}

void CompostR()
{
  for (_ABVAR_14_a=1; _ABVAR_14_a<= ( 60 ); ++_ABVAR_14_a ) // change (here)
  {
    __ardublockDigitalWrite(_ABVAR_3_Pusher, HIGH);
    delayMicroseconds( 500 );
    __ardublockDigitalWrite(_ABVAR_3_Pusher, LOW);
    delay( 20 );
  }
  _ABVAR_5_BinActive = 3 ;
}
```

```
void Recycling()
{
  for (_ABVAR_4_a=1; _ABVAR_4_a<= ( 20 ); ++_ABVAR_4_a )
  {
    __ardublockDigitalWrite(_ABVAR_2_Tilter, HIGH);
    delayMicroseconds( 1782 );
    __ardublockDigitalWrite(_ABVAR_2_Tilter, LOW);
    delay( 20 );
  }
  _ABVAR_5_BinActive = 2 ;
}
```

Improvements

If this was to be made for real life, it would be made out of stainless steel. The camera would be be able to detect more than just 7 colours, it would be able to recognise any rubbish from its colour, shape and size.

Acknowledgments

Hugh from Adelaide City Robotics Club helped me with coding the Pixy2 Camera. He helped me with the idea of how to move 2 cogs at once without having to have 2 servo motors running at the same time.

Heath from Adelaide City Robotics Club helped me when I had errors in my code. He taught me how to recalibrate the servos when they weren't moving properly.

Don from Adelaide City Robotics Club helped me work out what size cogs I would need by using multiplication.

Adelaide City Robotics Club members helped answer questions about my code whenever I was facing a problem.

Mummy A (Alisha) helped me by typing my write-up for me and taking me to Robotics Club each weekend to get help when I was stuck. She also helped me with my time management.

Nini (grandmother) took me to Robotics Club when Mummy A couldn't take me.