

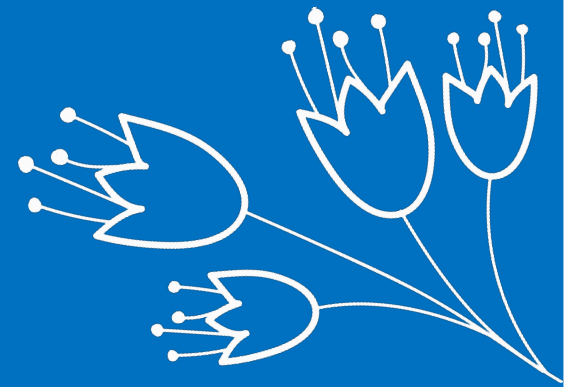
Intro to C++ programming



- ✓ Introduction
- ✓ About C++
- ✓ Data Types
- ✓ Loops
- ✓ Functions
- ✓ Arduino
- ✓ Advice

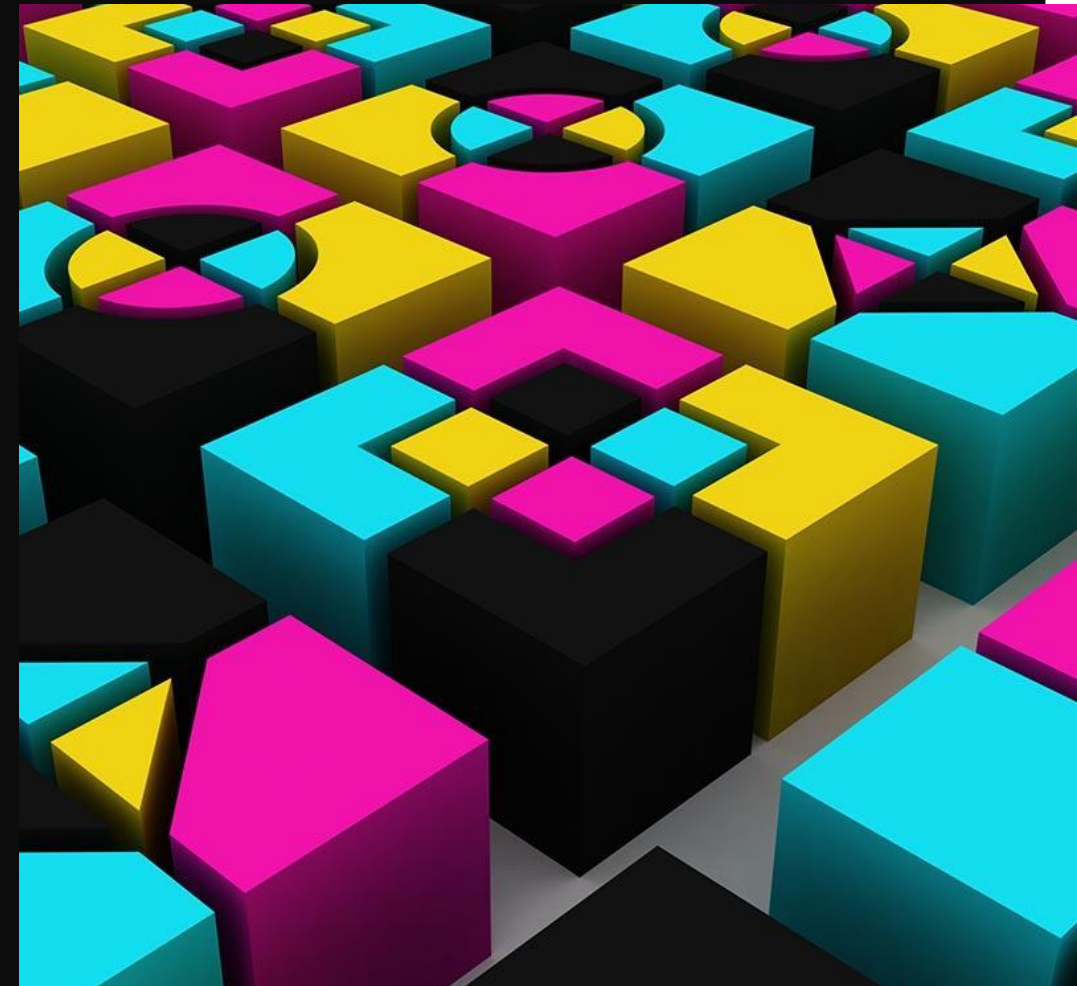


Presentation Main Points



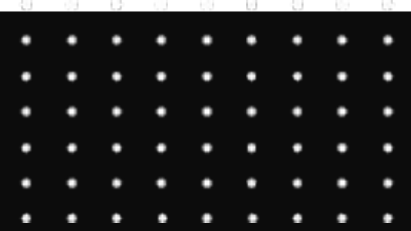
About C++

- Created for higher level of control over memory
- Object oriented programming
- Imperative language
- Compiled language

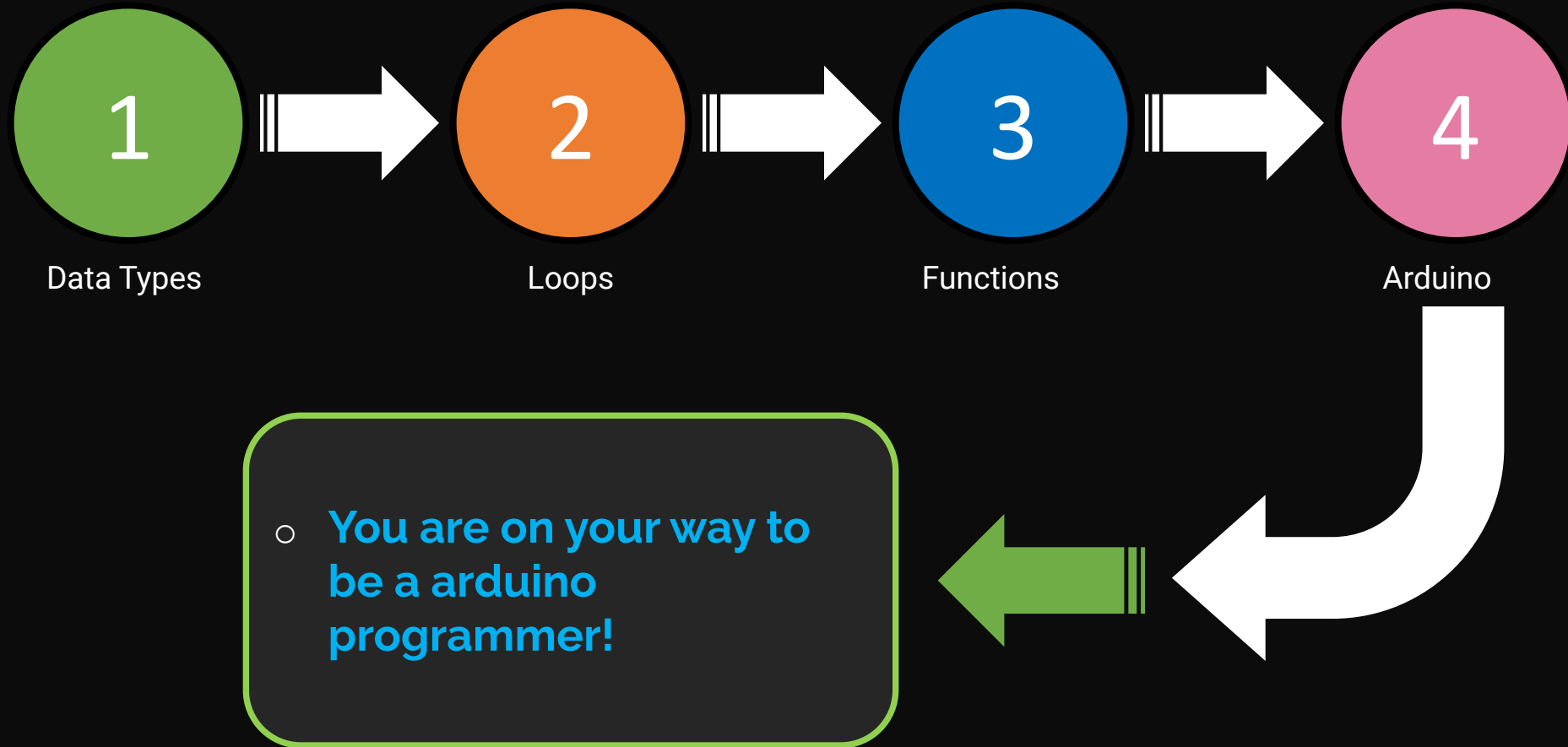


How much do you need to know?

- It rly depends
- On your project



Process Info-Graphic Slide





Python Data types

Strings

Integers/Numbers

Float

List, tuple

Dictionaries

boolean

Character - 1 byte

Boolean - 1 byte

Integer - 4 bytes

Float - 4 bytes

Double - 8 bytes

Long - 8 bytes

Arrays - 1 D, 2 D

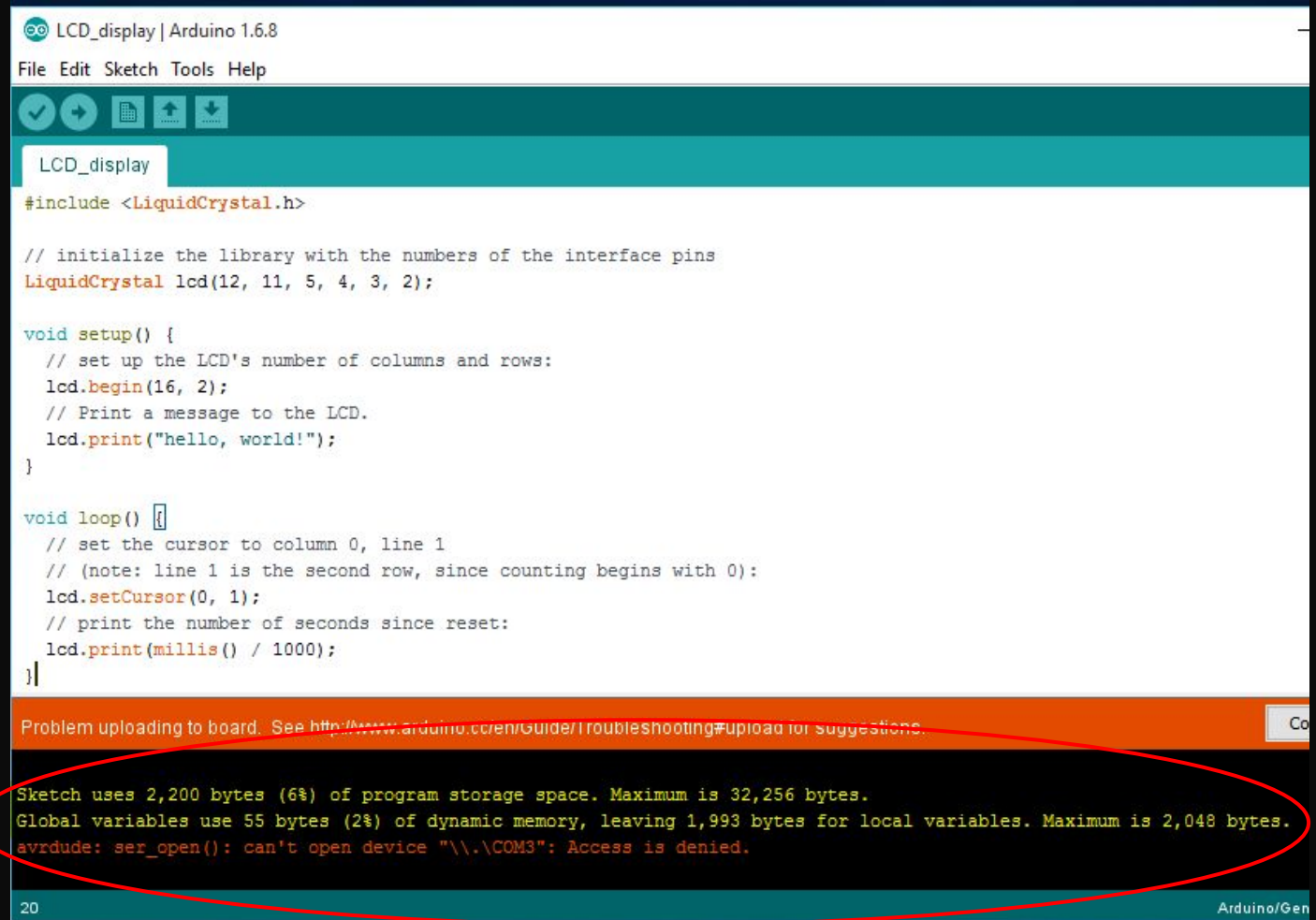
Many more

C++



Why memory?

- Potential memory problem later on
- Solved by
 - Connecting another arduino uno
 - Get a bigger arduino like mega



The screenshot shows the Arduino IDE interface with a sketch named 'LCD_display' for Arduino 1.6.8. The code includes the `LiquidCrystal` library and initializes an LCD with pins 12, 11, 5, 4, 3, and 2. The `setup` function sets the LCD to 16 columns and 2 rows and prints 'hello, world!'. The `loop` function sets the cursor to column 0, line 1 and prints the number of seconds since reset. An orange error bar at the bottom states: 'Problem uploading to board. See <http://www.arduino.cc/en/guide/troubleshooting#upload> for suggestions.' Below this, a red circle highlights a message box that says: 'Sketch uses 2,200 bytes (6%) of program storage space. Maximum is 32,256 bytes. Global variables use 55 bytes (2%) of dynamic memory, leaving 1,993 bytes for local variables. Maximum is 2,048 bytes. avrdude: ser_open(): can't open device '\\.\COM3': Access is denied.'

```
LCD_display | Arduino 1.6.8
File Edit Sketch Tools Help

LCD_display

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

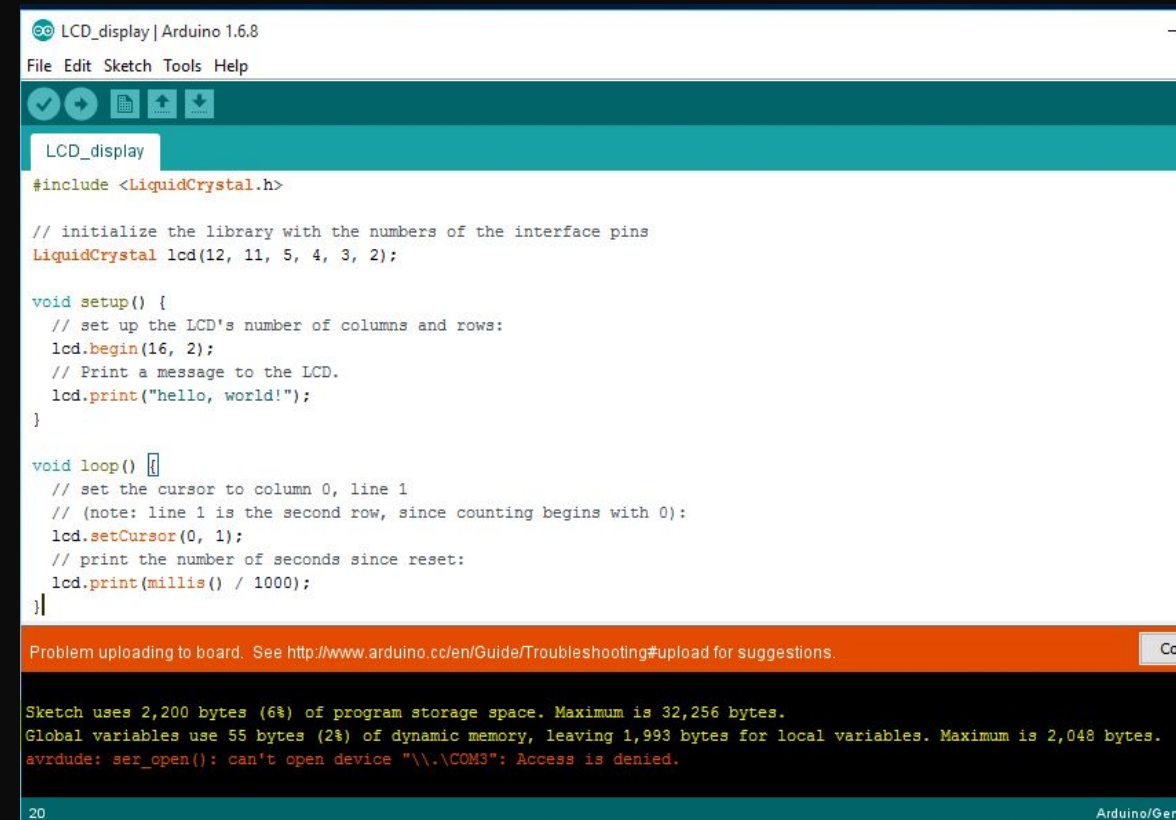
Problem uploading to board. See <http://www.arduino.cc/en/guide/troubleshooting#upload> for suggestions.

Sketch uses 2,200 bytes (6%) of program storage space. Maximum is 32,256 bytes.
Global variables use 55 bytes (2%) of dynamic memory, leaving 1,993 bytes for local variables. Maximum is 2,048 bytes.
avrdude: ser_open(): can't open device "\\.\COM3": Access is denied.

20 Arduino/Gen

How do we save memory?

- Memory allocation of variables
- Reduce print statements
- Lights -> brightness takes up mem
- Speaker -> volume takes up mem
- Try to estimate how much you need
- But have to consider cost
- If you are good can use optimization of your loops



```
LCD_display | Arduino 1.6.8
File Edit Sketch Tools Help

LCD_display
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

Problem uploading to board. See <http://www.arduino.cc/en/Guide/Troubleshooting#upload> for suggestions.

Sketch uses 2,200 bytes (6%) of program storage space. Maximum is 32,256 bytes.
Global variables use 55 bytes (2%) of dynamic memory, leaving 1,993 bytes for local variables. Maximum is 2,048 bytes.
avrdude: ser_open(): can't open device "\\.\\COM3": Access is denied.

20 Arduino/Gen



Python Data types

Declaring variables

Max_level = 10

A = []



C++

Declaring variables

```
const byte MAX_LEVEL = 10;
```

```
byte velocity = 1000;
```

```
bool soundonce = false;
```

```
byte sequence[MAX_LEVEL]; //c++ array number 10 sets 10 spaces,  
not set 10 as a variable inside.
```





Python loops

For loop
While loop
If else
Operators



C++

For loop
While loop
If else
Operators





Python loops

Declaring variables

x = 0

For i in range(0,10):

 x +=1



C++

Declaring variables

```
for (byte i = 0; i < 3; i++)
```

```
while ( generateseq == 1 )
```

```
if ( states ==0 ) {}
```

```
else if ( states == 1 ) {}
```

```
== || &&
```





Functions

Declaration

Def func1(arg):

Return



Functions

Declaration

```
void wrong_sequence() { ... }
```

```
void arduino2(int states) { ... }
```

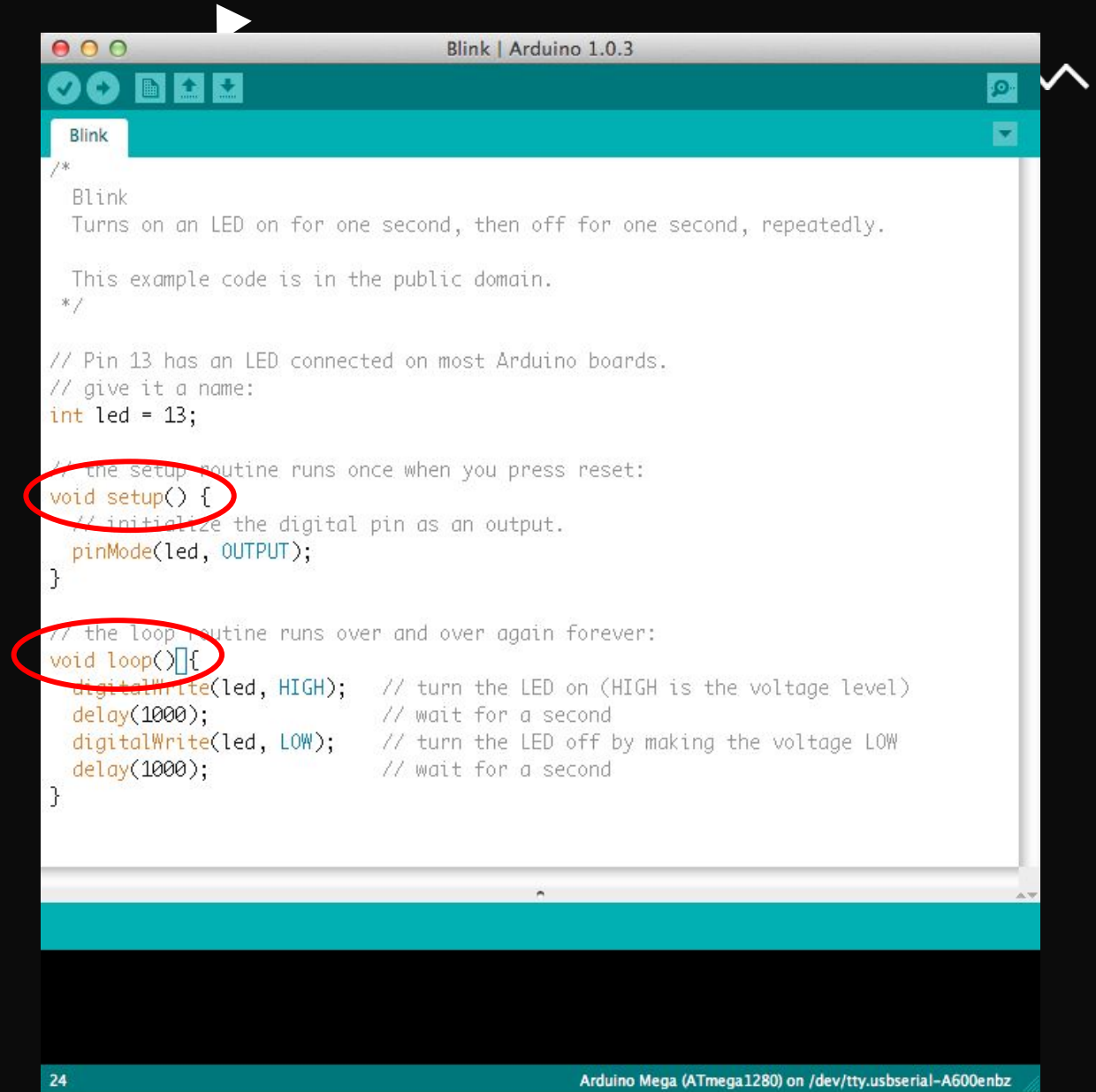


Arduino

- Setup()
- loop()
- pinMode()
- delay()
- digitalRead()
- digitalWrite()

red led pin number 7

Green led pin number 13



```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

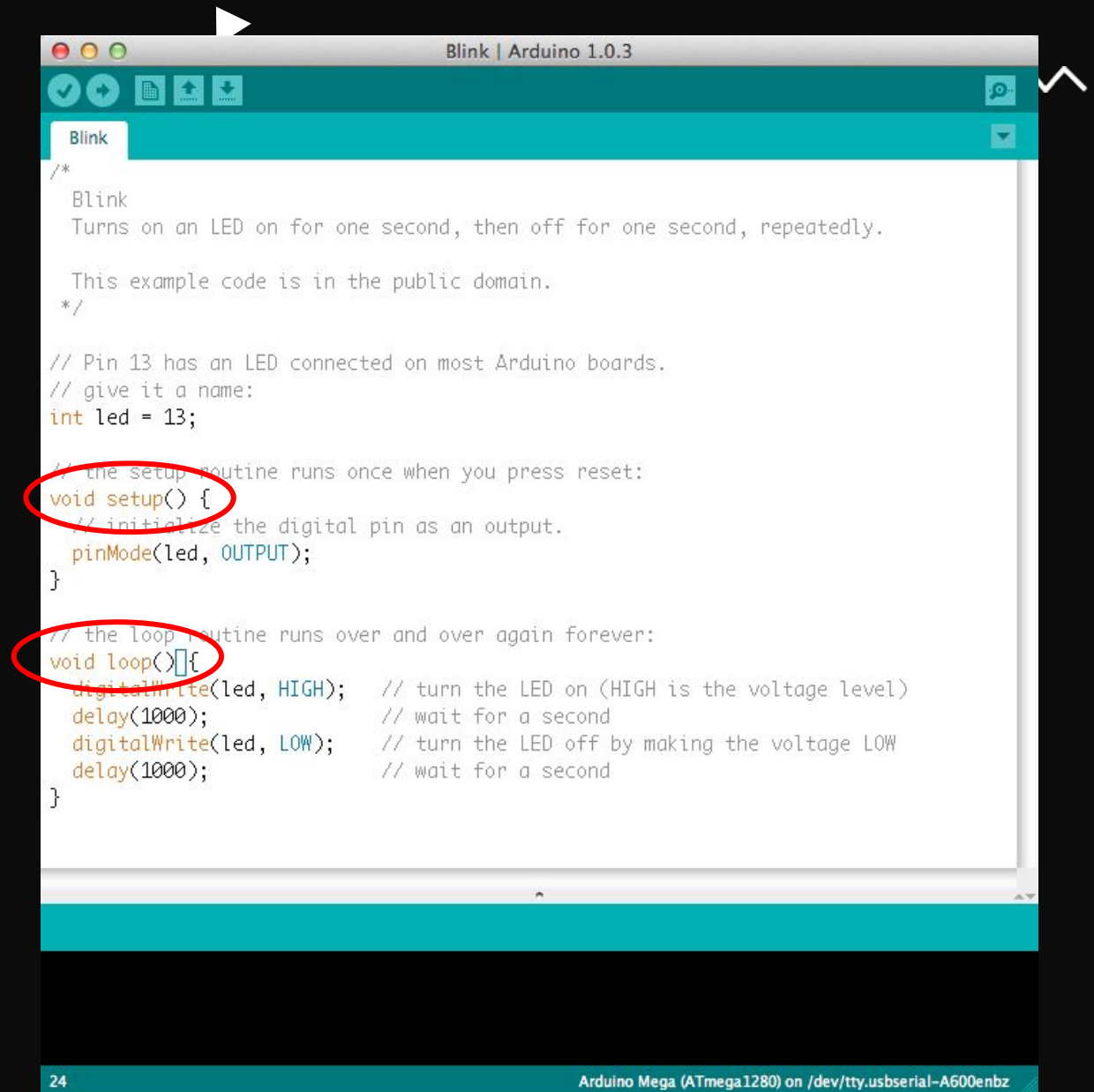
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

24 Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz

Arduino Activity

Make one 1 led blink (On and off)

1. Define your pin number
2. Inside Setup() use pinMode()
3. Inside loop() use digitalWrite()



```
Blink | Arduino 1.0.3

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

24 Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz

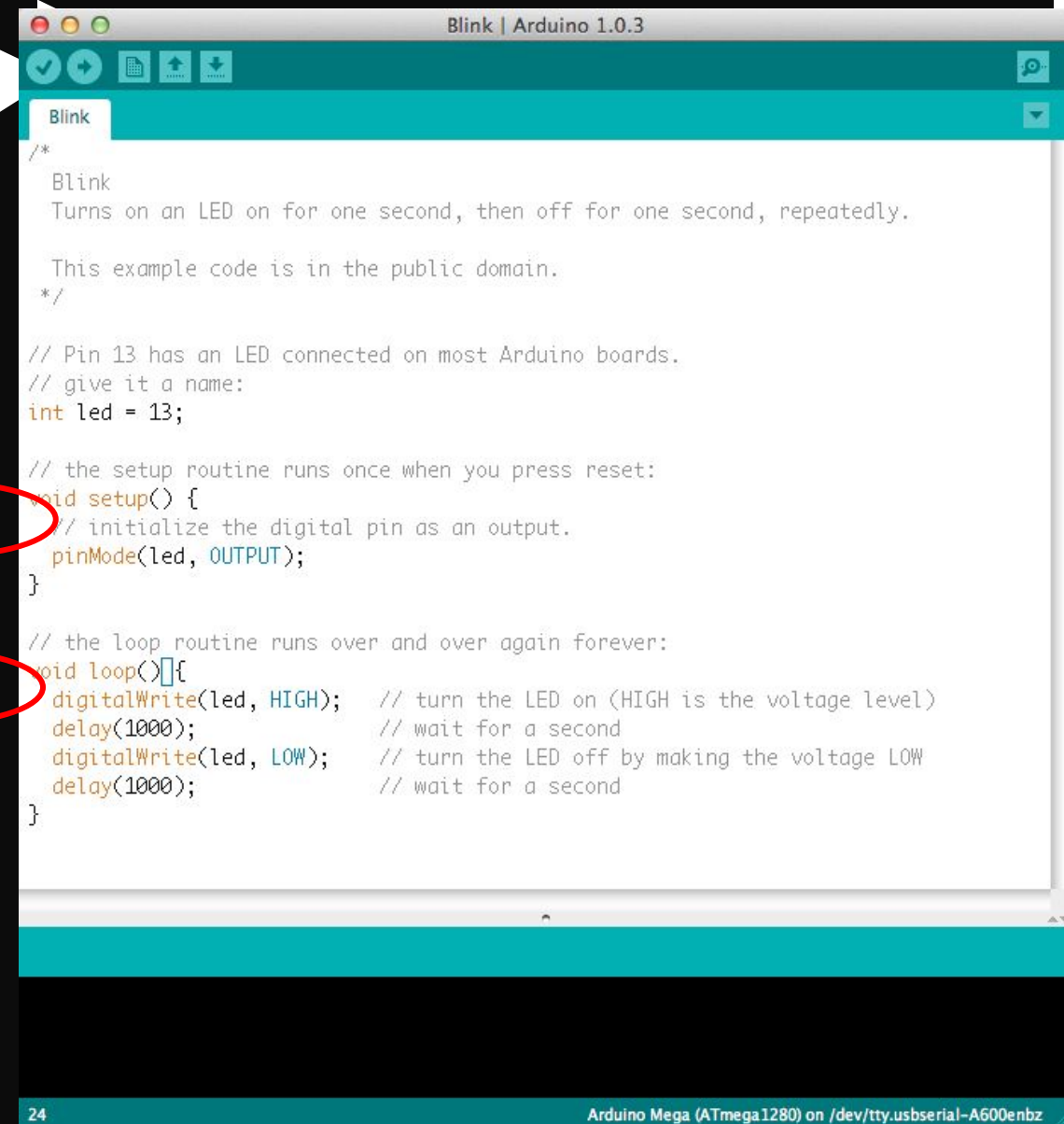
Arduino Activity 2

red led pin number 7

Green led pin number 13

Given 2 leds make them blink simultaneously

[HOW TO MAKE 2 LEDS BLINK AT THE SAME TIME? - Using Arduino / Project Guidance - Arduino Forum](#)



```
Blink | Arduino 1.0.3

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

24 Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz



Advice

- Start Early -> plan your project solution -> go through the details, what parts to buy, what code do you need to do, how much memory do you need.
- If i knew a speaker and a 5m 5V LED strip required an arduino mega i would have gotten a mega instead of a uno.
- Ask advice from seniors/people who are good at certain areas if your idea is feasible and doable.
- Check google, stackoverflow, arduino blog
- Use tinkercad



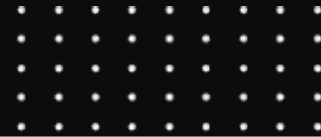
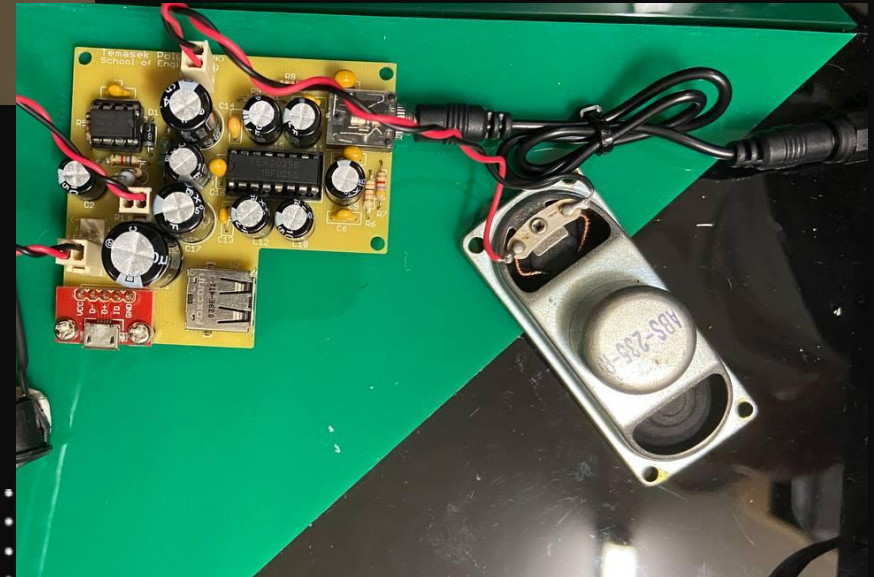
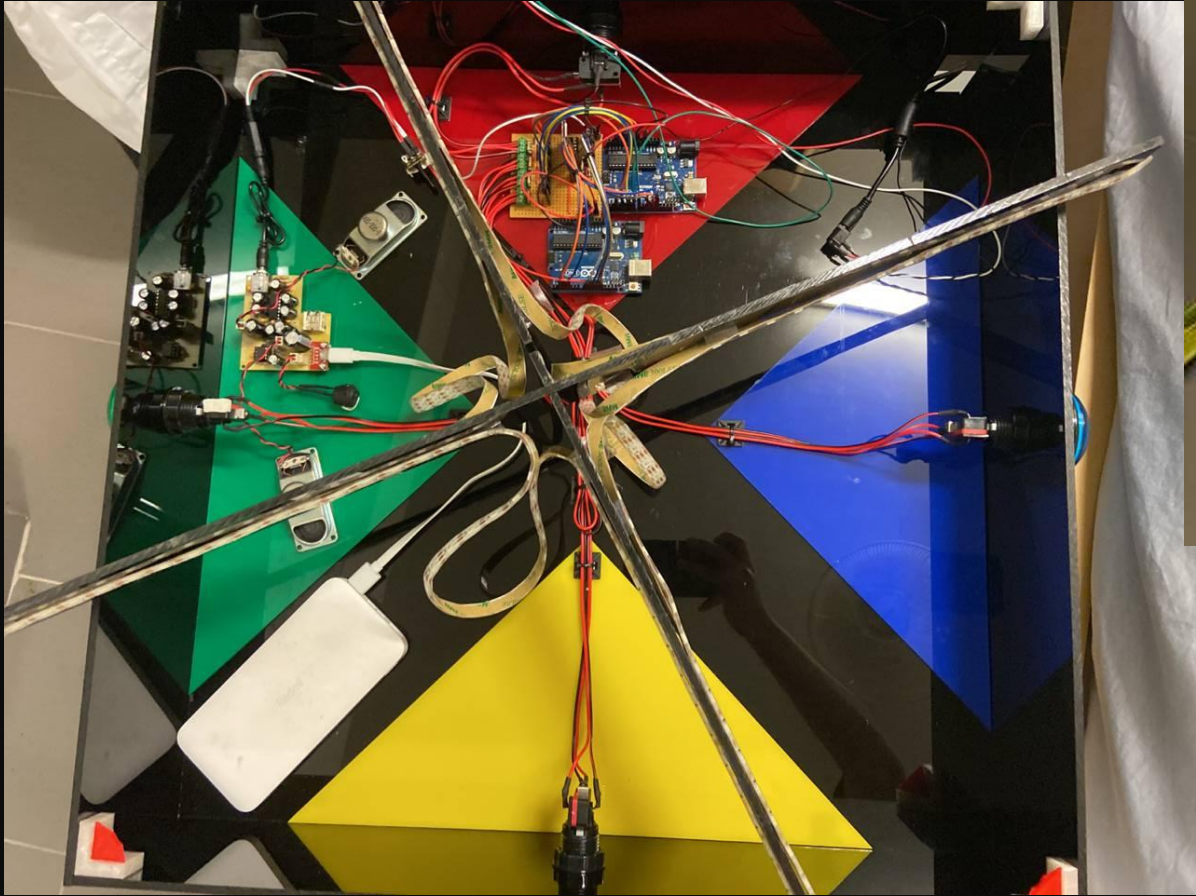
Ideas

- My project
- Multi-functional table
 - Game table
 - Normal table
- What did it require?
- Coding side c++ knowledge
- 2 Arduino unos
- Multi-threading
- Memory allocation
- Speaker and lights library
- Binary numbers

[C5G1 Project Part 4 – 3.007 Design Projects \(sutd.edu.sg\)](http://sutd.edu.sg)



Ideas



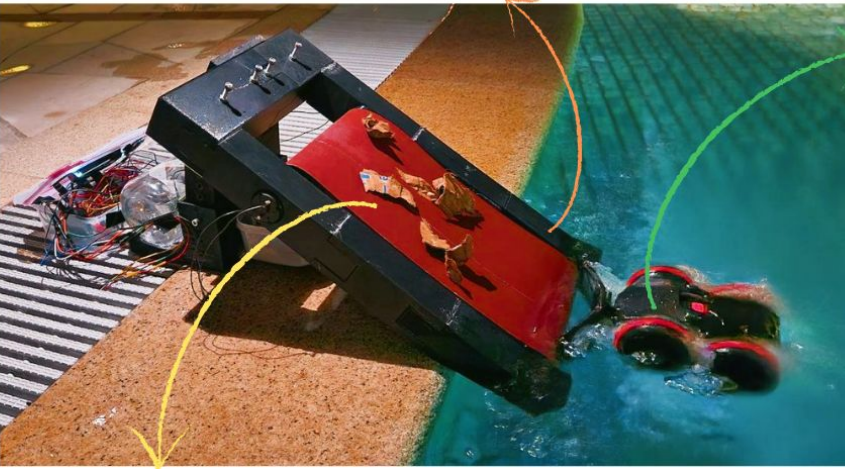
Ideas

- YongLiang's project
- Floating Robot game
- What did it require?
- Coding side c++ knowledge
- Arduino mega
- Weight sensor and bluetooth modules
- Lcd and game screens
- Scoreboard
- Firebase

[Remote Controlled Koi \(RCK\) – 3.007 Design Projects \(sutd.edu.sg\)](https://sutd.edu.sg)

- It is a thermoplastic and can be cooled back to appropriate temperature without any significant degradation.
- It is non-toxic and can be placed near the pond without causing any harm to the surroundings.

- It has a reduced carbon footprint, as compared to fossil-based plastics as it is made from plants and required less energy to be produced.
- It requires approximately **1.59MJ** to produce 1kg of it.



Amphibious Robot

MATERIAL CHOICE

Polyolefins - Polypropylene

- Polypropylene is very strong and thus is able to withstand any collisions if the user is playing with the robot
- It is also highly heat resistant, suitable for the hot weather.
- It is very lightweight and will not sink into the pond or be disruptive to the flora or fauna.

Polyolefins - Polyethylene

- Polyethylene is impact resistance which will protect it from any damage during games.
- It is also corrosion resistant and will not experience oxidation or other natural chemical reactions.

Conveyor Belt

Sandpaper - MATERIAL CHOICE

- Sandpaper is very rough and can bring trash/leaves that it carries up the slope without slipping.
- It is able to withstand a lot of weight without

ENVIRONMENTAL IMPACT

Polyolefins - Polypropylene

- Polypropylene is desirable as it can be recycled many times and it not a big environmental concern in usage.
- However, it consumes high amounts of water and energy to manufacture.
- It is manufactured using petroleum and natural gases which emit large amounts of greenhouse gases during



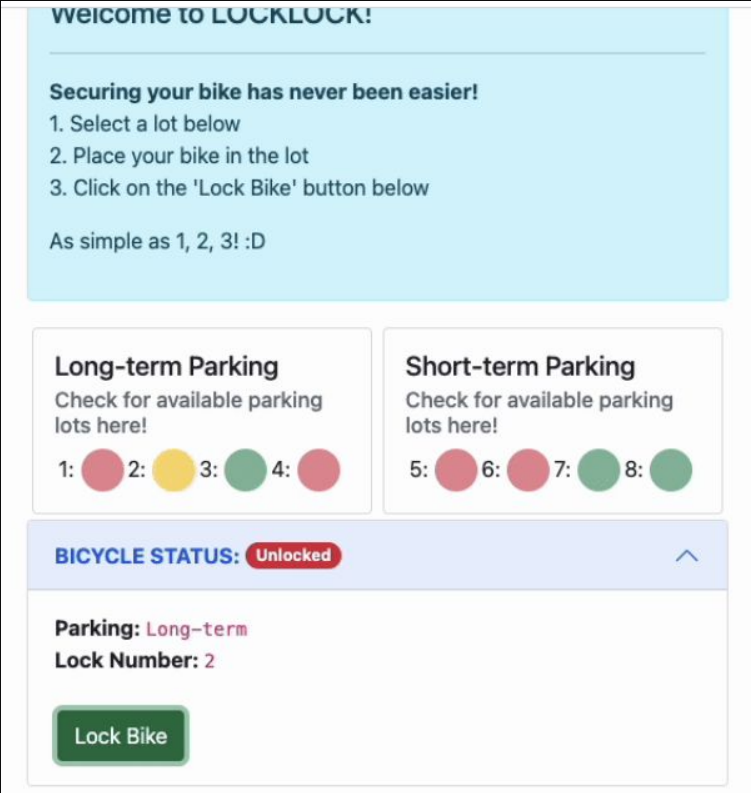
Ideas



Ideas

- Beckham and Fernanda's project
- Bicycle Lock
- What did it require?
- Coding side c++ knowledge
- Arduino uno
- Wifi modules and firebase module
- Web application - firebase and django framework (html, css, js)

[SC04 Group 5 Part 4 – 3.007 Design Projects \(sutd.edu.sg\)](#)



The screenshot shows a web application for a bicycle lock system. At the top, a light blue banner says "welcome to LOCKLOCK!". Below this, a section titled "Securing your bike has never been easier!" lists three steps: 1. Select a lot below, 2. Place your bike in the lot, and 3. Click on the 'Lock Bike' button below. It then says "As simple as 1, 2, 3! :D".

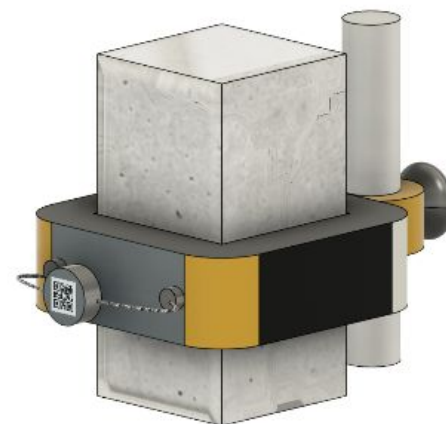
Below the banner are two panels for parking lots. The "Long-term Parking" panel says "Check for available parking lots here!" and shows four colored circles: 1 (red), 2 (yellow), 3 (green), and 4 (red). The "Short-term Parking" panel says "Check for available parking lots here!" and shows four colored circles: 5 (red), 6 (red), 7 (green), and 8 (green).

Below these panels is a light blue bar with the text "BICYCLE STATUS: Unlocked" and a small upward arrow icon. Below this bar, the text "Parking: Long-term" and "Lock Number: 2" is displayed. At the bottom of the interface is a green button labeled "Lock Bike".

Ideas



2) Height adjustable bicycle lock attached to the poles





Telegram handle

- @ShjonathanTan - Shjonathan
- @lamthebob - Billy
- @YL_SB - Yong liang

- @designacademysutd - channel

