

A decorative graphic on the left side of the slide, consisting of a network of white lines and circles on a blue background, resembling a circuit board or a tree structure.

CHARGING BIKE STATION SECOND GENERATION

ELECTRICAL SYSTEMS

ALHARBI, DEVINE, DHILLON

10/30/13

OVERVIEW

- Problem Statement
- Gantt chart
- Concept Generation
- Languages
- Chipset
- Generators
- Concept Selection
- Summary
- References

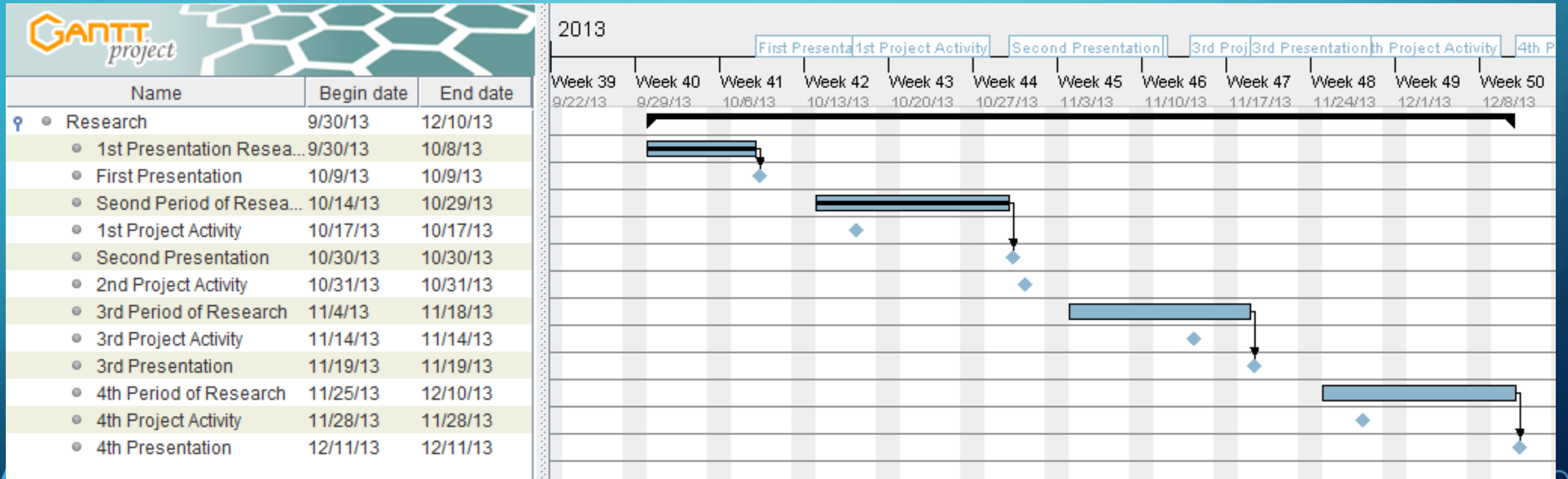
PROBLEM STATEMENT

- The project is to develop a bicycle that will generate mechanical power from a user to various electronic devices and that will have an aesthetically looking display that displays unique parameters of the individual powering their devices.

PROBLEM STATEMENT CONT.

- These parameters are the total power consumption of the mechanical power being consumed at the time of the individual using the bike and the total power consumed in a lifetime of the bicycle.

GANTT CHART



CONCEPT GENERATION

- Choose a programming language to run the graphical user interface
- Choose a chipset to run the program and display
- Choose a generator style to run the system
- Choose to use batteries or capacitors
- Choose what kind of display to use

LANGUAGES

- C
- Python
- VHDL
- Verilog

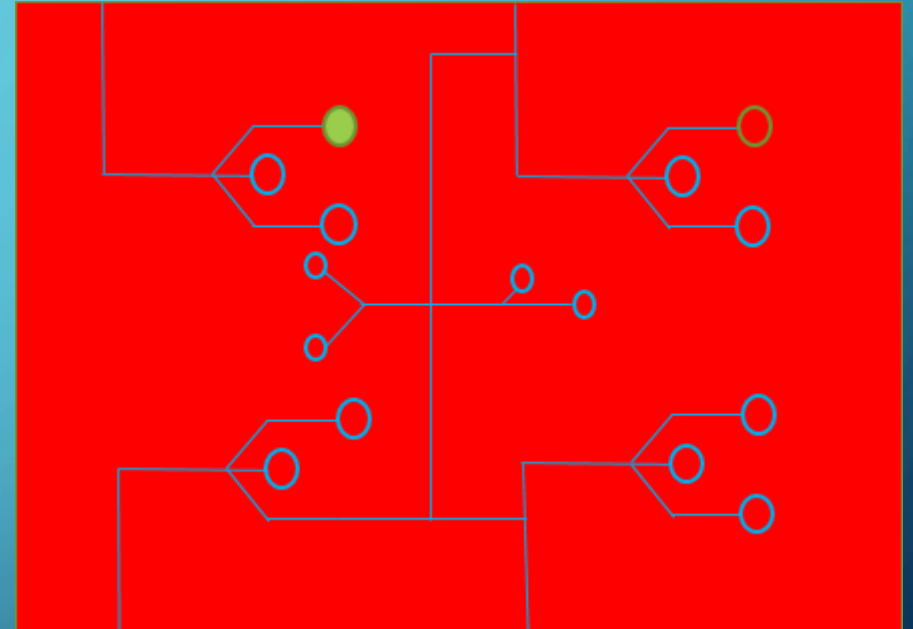
PROGRAMMING IN C

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

From StackOverFlow Forums



ADVANTAGES OF C

- Very large install base
 - Many useful tools and support available
- Has graphics libraries
 - Can display complex graphics
- Supported by nearly all devices
 - Will work on any chipset we decide to use

DISADVANTAGES OF C

- Complicated programming language
 - Takes longer to program and debug
- Not Hardware
 - Does not run as fast

PROGRAMMING IN PYTHON

```
import RPi.GPIO as GPIO ## Import GPIO library
import time ## Import 'time' library. Allows us to use 'sleep'

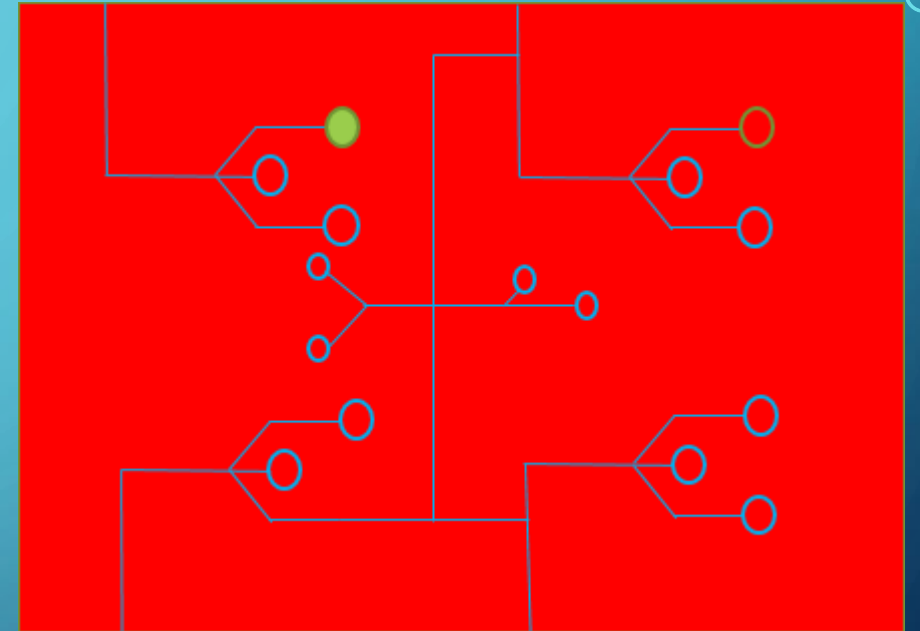
GPIO.setmode(GPIO.BOARD) ## Use board pin numbering
GPIO.setup(7, GPIO.OUT) ## Setup GPIO Pin 7 to OUT

##Define a function named Blink()
def Blink(numTimes,speed):
    for i in range(0,numTimes):## Run loop numTimes
        print "Iteration " + str(i+1)## Print current loop
        GPIO.output(7,True)## Switch on pin 7
        time.sleep(speed)## Wait
        GPIO.output(7,False)## Switch off pin 7
        time.sleep(speed)## Wait
    print "Done" ## When loop is complete, print "Done"
    GPIO.cleanup()

## Ask user for total number of blinks and length of each blink
iterations = raw_input("Enter total number of times to blink: ")
speed = raw_input("Enter length of each blink(seconds): ")

## Start Blink() function. Convert user input from strings to
numeric data types and pass to Blink() as parameters
Blink(int(iterations),float(speed))
```

From ThirdEyeVisions Tutorials



ADVANTAGES OF PYTHON

- Loosely typed scripting language
 - Easy to develop
- Has graphics libraries
 - Can display complex graphics
- Can be compiled to machine code
 - Can be used on most embedded devices

DISADVANTAGES OF PYTHON

- Scripting language
 - Runs slower than a compiled language
- Loose syntax
 - Bugs are harder to identify and correct
- Small install base
 - Fewer tools and support available

PROGRAMMING IN VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

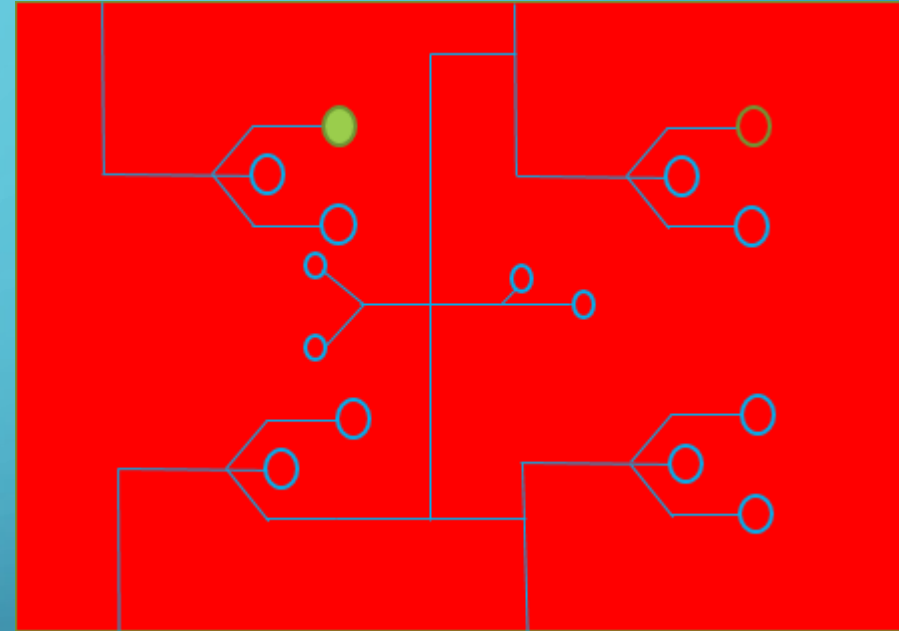
entity Clk_div_led is
  Port (
    Clk          : in  std_logic;
    led_cathode  : out std_logic;
    led_anode    : out std_logic
  );
end Clk_div_led;

architecture RTL of Clk_div_led is
  constant max_count : natural := 48000000;
  signal Rst_n       : std_logic;
begin

  Rst_n <= '1';
  led_cathode <= '0';

  -- 0 to max_count counter
  compteur : process(Clk, Rst_n)
    variable count : natural range 0 to max_count;
  begin
    if Rst_n = '0' then
      count := 0;
      led_anode <= '1';
    elsif rising_edge(Clk) then
      if count < max_count/2 then
        led_anode <='1';
        count := count + 1;
      elsif count < max_count then
        led_anode <='0';
        count := count + 1;
      else
        count := 0;
        led_anode <='1';
      end if;
    end if;
  end process compteur;
end RTL;
```

From ArmadeuS Project



ADVANTAGES OF VHDL

- Hardware definition language
 - Runs very fast
- Portable design
 - Can be compiled to run on different devices
- Concurrent dataflow
 - Can do multiple operations concurrently

DISADVANTAGES OF VHDL

- No graphics library
 - All graphics would need to be programmed by hand
- Small install base
 - Fewer tools and support available
- Verbose
 - Programs are lengthy and confusing

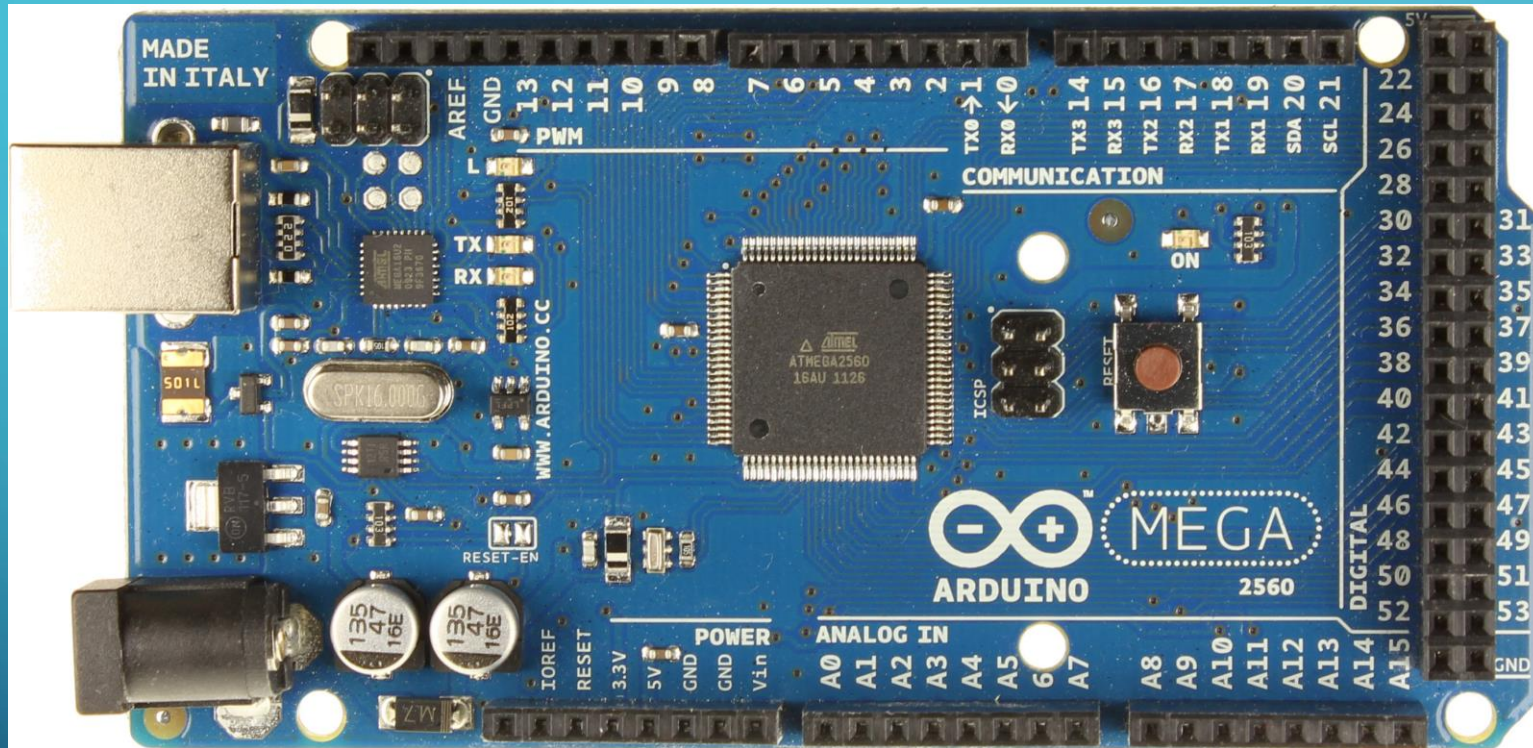
LANGUAGE DECISION MATRIX

Factor	Factor Weight	C	Python	VHDL
Performance	20	7	5	9
Utilities	20	9	7	2
Graphics	50	9	9	2
Ease	10	5	9	3
Total	100			
Ranking	Score			
Python	7.80			
C	6.04			
VHDL	3.50			

CHIPSETS

- Arduino
- MSP430
- Raspberry Pi

ARDUINO MICROPROCESSORS

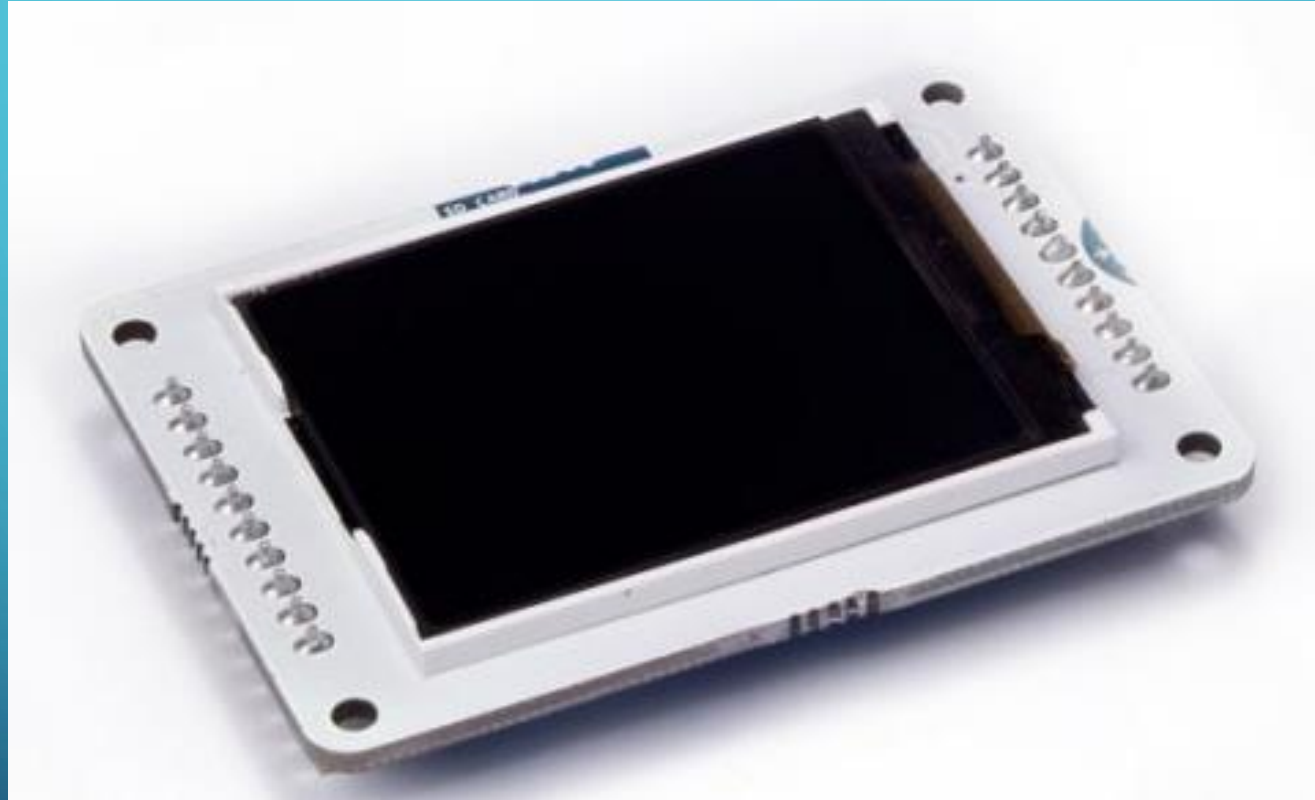


From Arduino

ADVANTAGES OF ARDUINO

- Has dedicated display output
 - Purpose built to use high definition displays
- Modular components
 - Can build hardware for a specific purpose
- Simple programming environment
 - Easy to implement programs
- Integrated Display
 - Can have a built in high definition display

DISPLAY

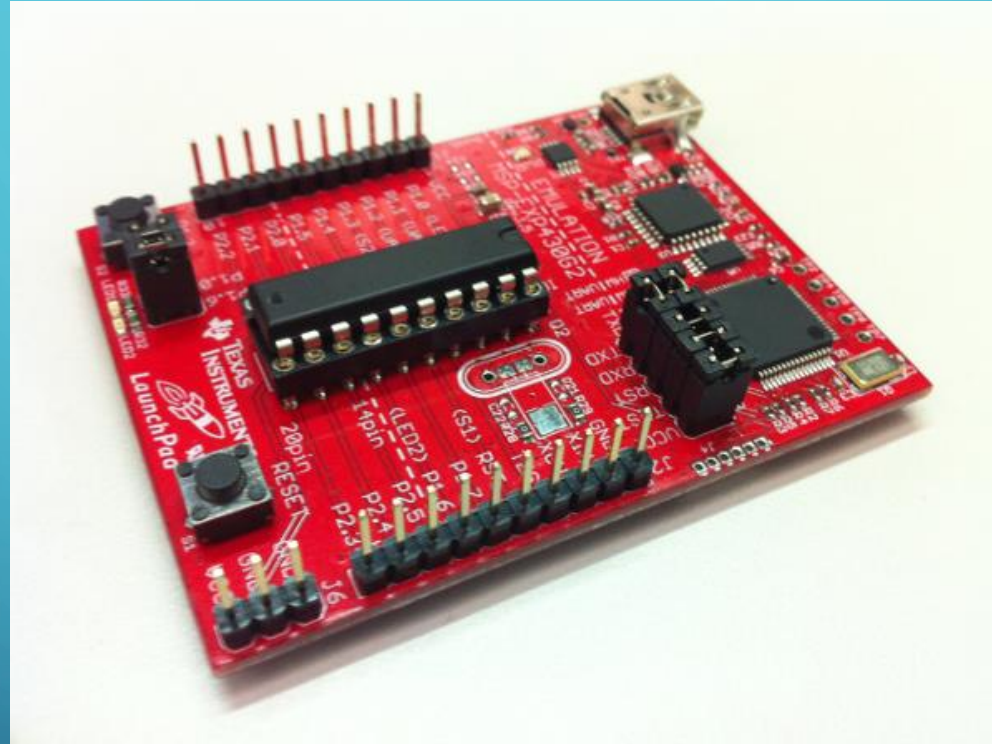


From Arduino store

DISADVANTAGES OF ARDUINO

- Low hardware specifications
 - Limited to small scale applications
- ARM processor
 - Slower speeds

MSP430



From Texas Instruments

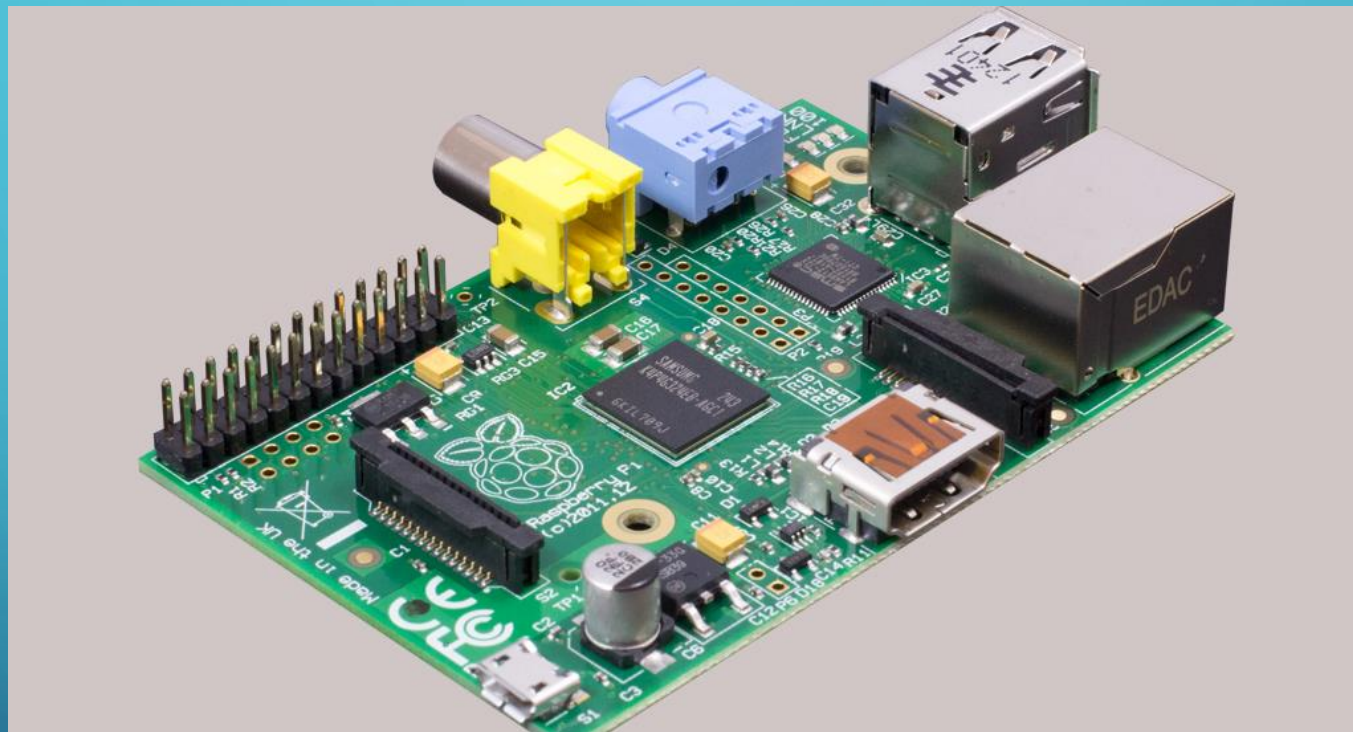
ADVANTAGES OF MSP430

- Many different variations
 - Can choose hardware based on requirements
- Low power consumption
 - More power to charge the device

DISADVANTAGES OF MSP430

- Limited memory
 - Cannot store as much data
- No external bus
 - Can only be used in very simple applications

RASPBERRY PI



From Raspberry Pi

ADVANTAGES OF RASPBERRY PI

- Extendable Memory
 - Can store very large amounts of data
- Built in Linux operating system
 - Programs can run on top of existing architecture
- Multiple Expansion ports
 - Can keep track of multiple sensors at once

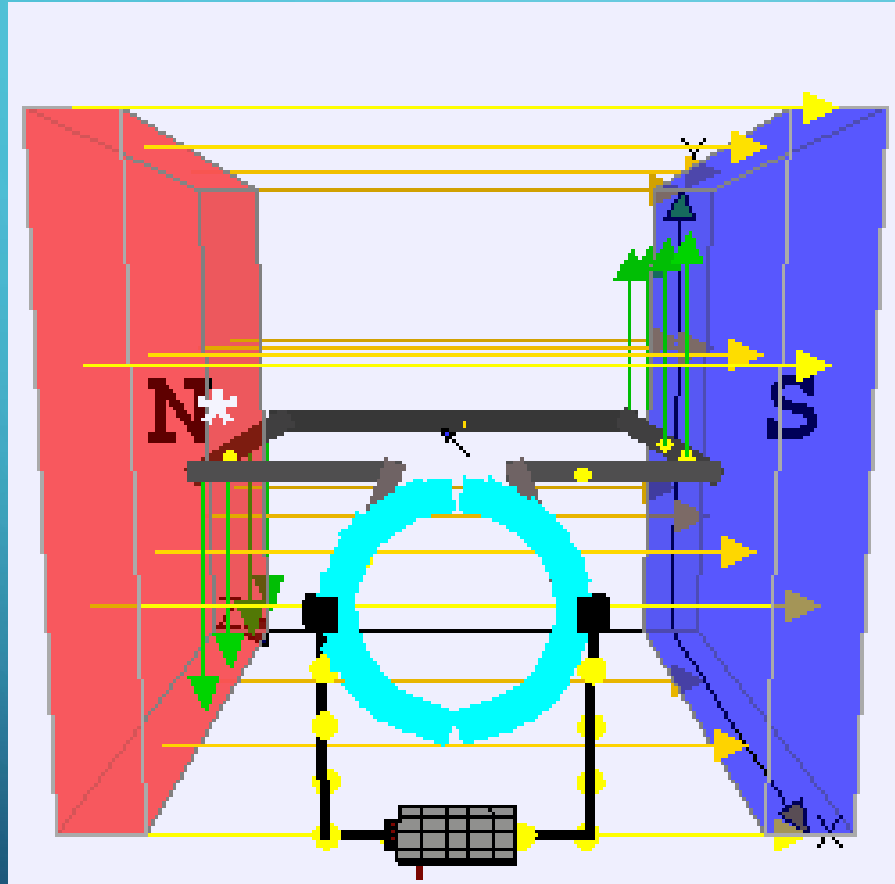
DISADVANTAGES OF RASPBERRY PI

- Only one model
 - Hardware is predefined
- Higher power consumption
 - Less power to charge the device

CHIPSET DECISION MATRIX

Factor	Factor Weight	Arduino	MSP430	Raspberry Pi
Performance	5	7	7	9
Memory	30	7	3	9
Graphics	50	5	7	9
Customizable	15	7	9	2
Total	100			
Ranking	Score			
Raspberry Pi	7.95			
Arduino	7.00			
MSP430	6.10			

GENERATOR



From Wikipedia

- Magnetic field lines
- Magnets
- Coil
- Load

GENERATORS

- In wheel purchased generator
- External purchased generator
- Custom built generator

IN WHEEL GENERATOR



From Rock the Bike

ADVANTAGES OF IN WHEEL

- Highly Portable
 - Directly connected to bike
- Shorter Connections
 - Less resistance

DISADVANTAGES OF IN WHEEL

- Expensive
 - Less money to use on other parts
- Made to be ridden on a road
 - Less power output due to less resistance

EXTERNAL PURCHASED GENERATOR



From Testically

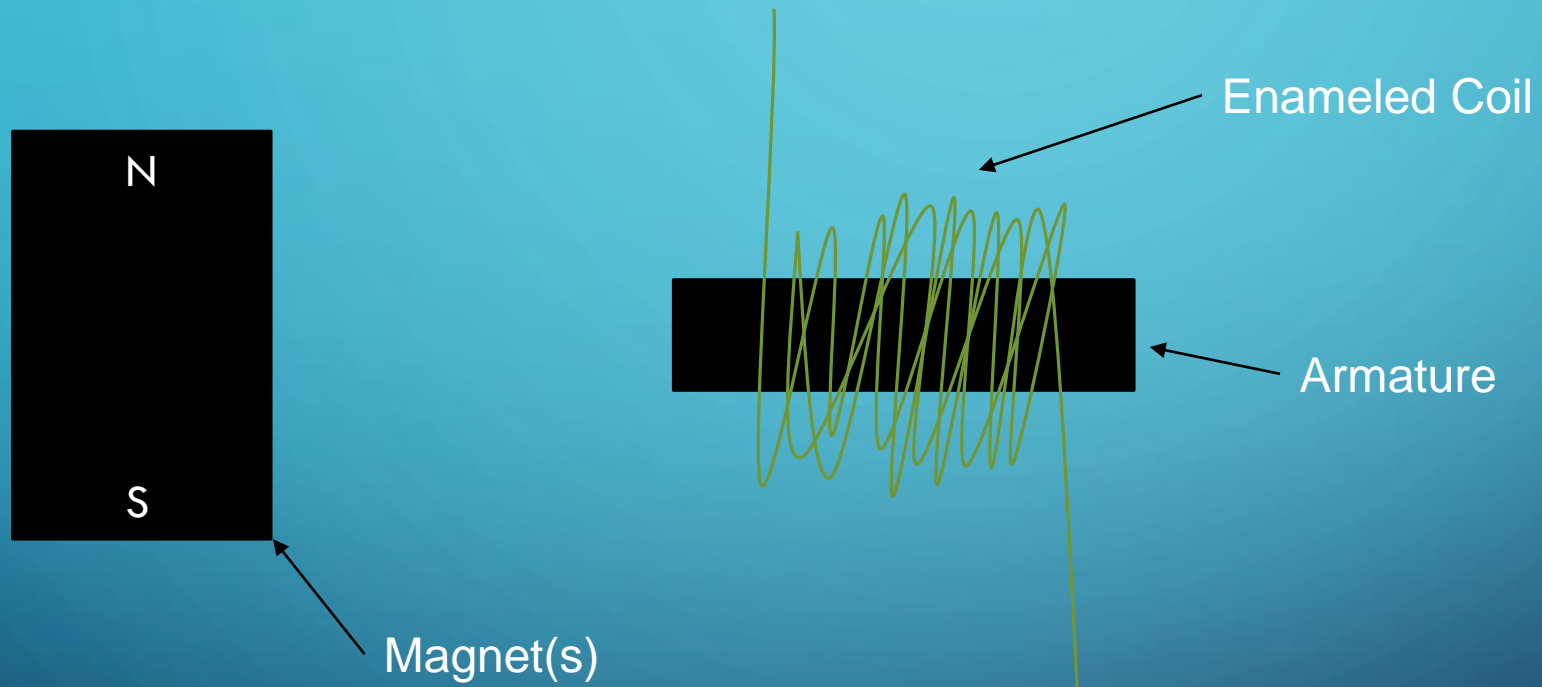
ADVANTAGES OF EXTERNAL

- Premade
 - Already optimized
- Reliable
 - Lower chance of failure

DISADVANTAGES OF EXTERNAL

- Expensive
 - Less money to use on other parts
- Bulky
 - Difficult to design around

CUSTOM BUILT GENERATOR



ADVANTAGES OF CUSTOM

- Inexpensive
 - More money to spend on other parts
- Open box
 - Easier to fix failures
- Customized Design
 - Built to fit in the mechanical team's design

DISADVANTAGES OF CUSTOM

- Hand made
 - Not as reliable
- Complex
 - Takes a lot of time to make

GENERATOR DECISION MATRIX

Factor	Factor Weight	In Wheel	External	Custom
Efficiency	25	7	9	6
Portability	25	9	4	9
Reliability	25	7	9	5
Low Cost	25	3	5	9
Total	100			
Ranking	Score			
Custom	7.25			
External	6.75			
In Wheel	6.50			

CONCEPT SELECTION

- Programming Languages
 - Python
- Chipsets
 - Raspberry Pi
- Generators
 - Custom Generator

IMPORTANCE DECISION MATRIX

Factor	Factor Weight	Generator	Display	Programming Language	Chip
Speed	5	7	9	7	7
Aesthetics	45	7	9	5	5
Display					
Parameters	25	8	7	9	9
Stability	10	8	9	7	7
Performance	15	9	9	9	7
Total	100				
Ranking	Score				
Display	850				
Generator	765				
Programming Language	690				
Chip	660				

The background is a solid teal color. In the four corners, there are decorative white line-art patterns resembling circuit traces or neural network connections. These patterns consist of thin lines that branch out and terminate in small circles.

QUESTIONS?

REFERENCES

- StackOverFlow forums
 - <http://stackoverflow.com/questions/78744/how-to-start-programming-a-microcontroller>
- SparkFun
 - <http://www.sparkfun.com/products/10914>
- Arduino Store
 - store.arduino.cc/ww/index.php
- Wikipedia
 - http://en.wikipedia.org/wiki/DC_motor
- ThirdEyeVisions
 - <http://www.thirdeyevis.com/pi-page-2.php>
- ArmadeuS
 - http://www.armadeus.com/wiki/index.php?title=Simple_blinking_LED
- Texas Instruments
 - http://www.ti.com/lscs/ti/microcontroller/16-bit_msp430/overview.page
- Raspberry Pi
 - <http://www.raspberrypi.org/>
- Rock the Bike
 - <http://rockthebike.com/store/224-thickbox/handbuilt-24-hub-generator-wheel.jpg>
- Testically
 - <http://www.testically.org/wp-content/uploads/2011/02/generator.jpg>