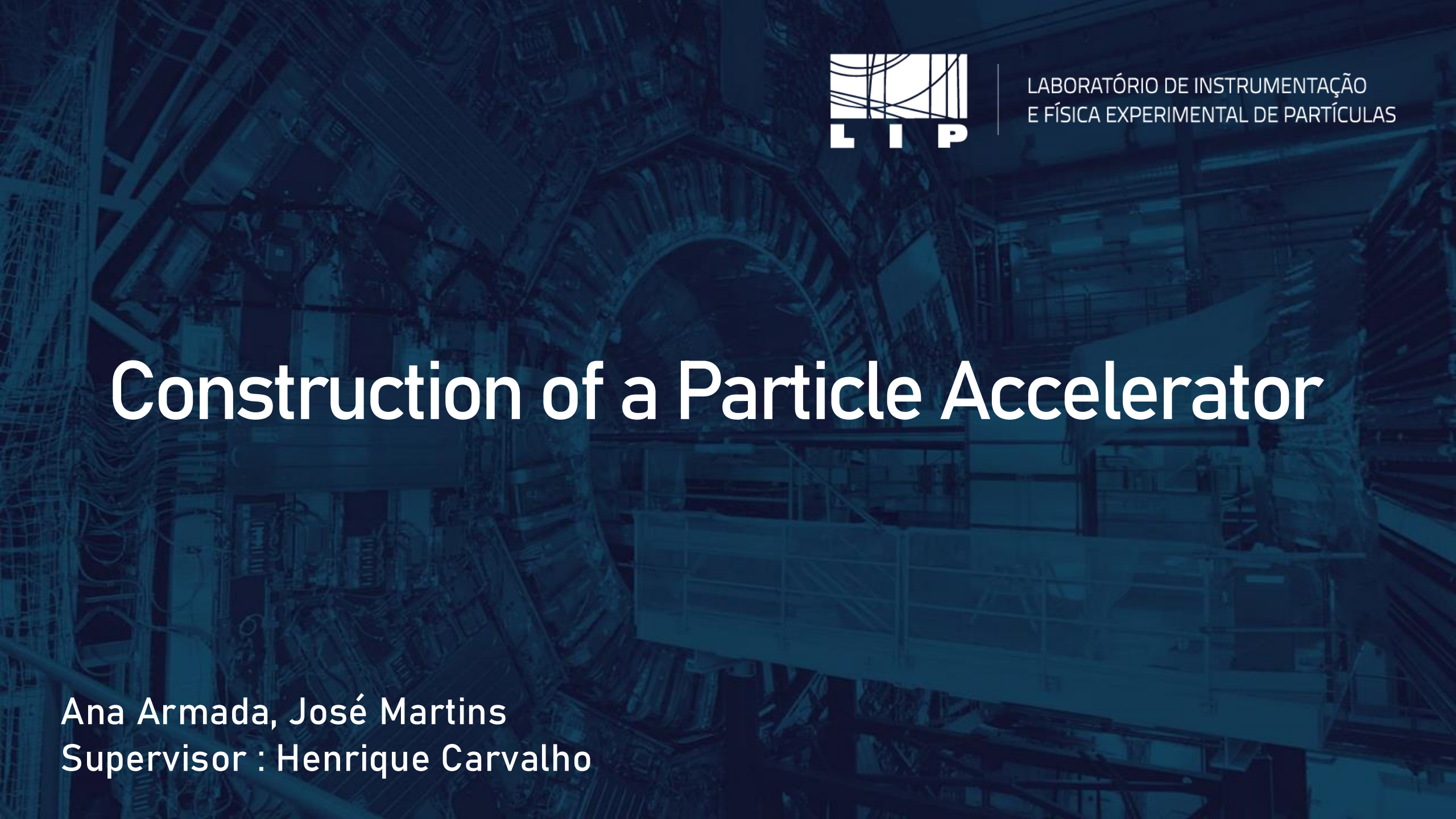


LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS

Construction of a Particle Accelerator

Ana Armada, José Martins
Supervisor : Henrique Carvalho

The LHC

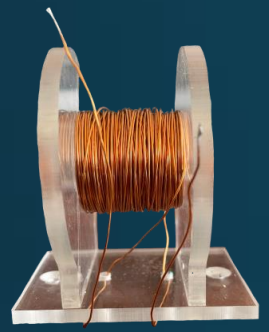
- The Large Hadron Collider (LHC) is the world's biggest and most powerful particle accelerator;
- Collide locations (ATLAS, CMS, ALICE, LHCb);
- 27 km.

How does it Work

Vacuum system
Beam Pipes
Magnets
Helium line

Magnets
Lattice Magnets
Insertion Magnets
Dipole Magnets

From the LHC to our project



Our little Booster

- 150 turns of 5 mm copper wire
- 0,0348 T (4A)

The Button

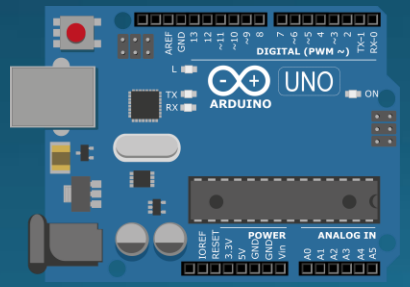
- Click threshold

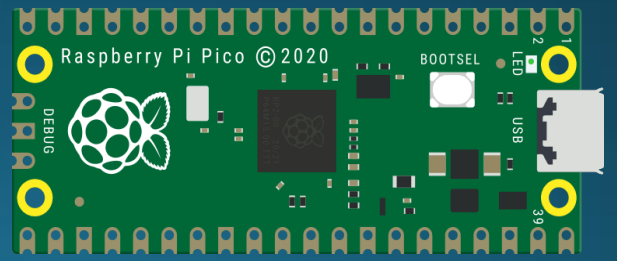
- IR emitter and receiver
- RPM function
- RPM light sensor

Microcontroller

Raspberry Pi Pico

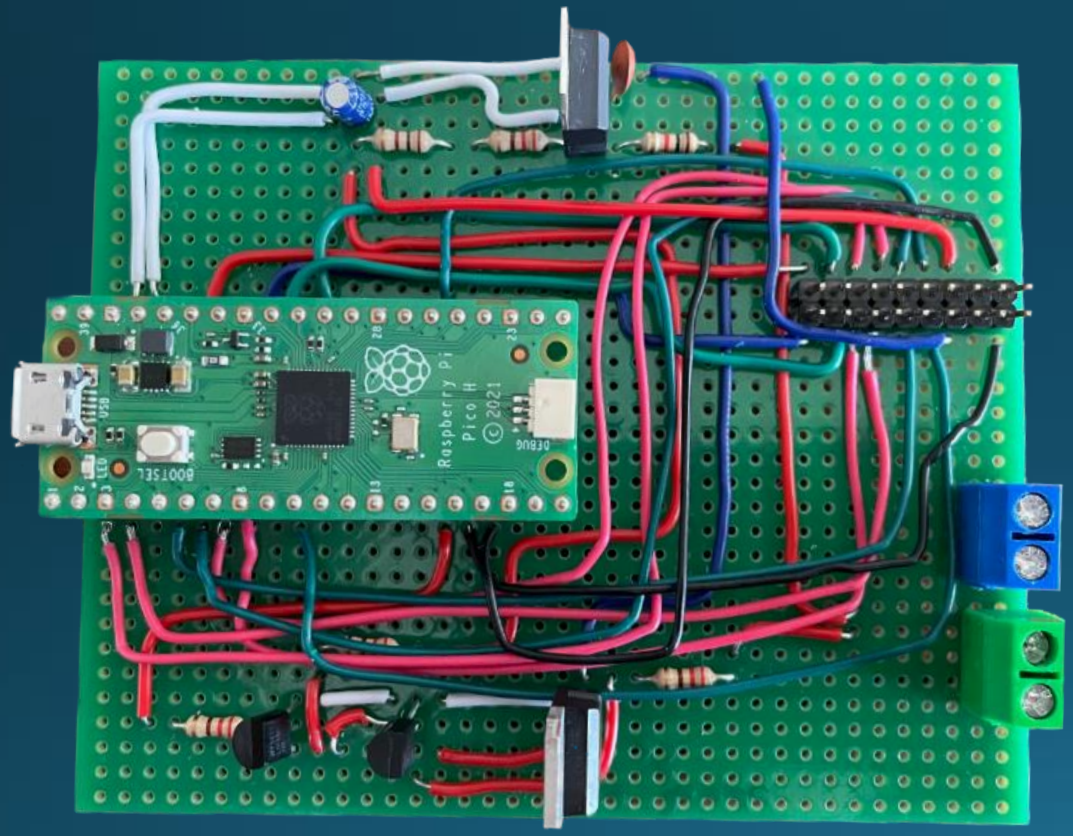
SPECS	ARDUINO UNO	RASPBERRY PI PICO
CORE	SINGLE-CORE	DUAL-CORE
RAM	2 KB	264 KB
CLOCK	16 MHZ	48 MHZ
FLASH MEMORY	32 KB	2 MB
OUTPUTS	20	26

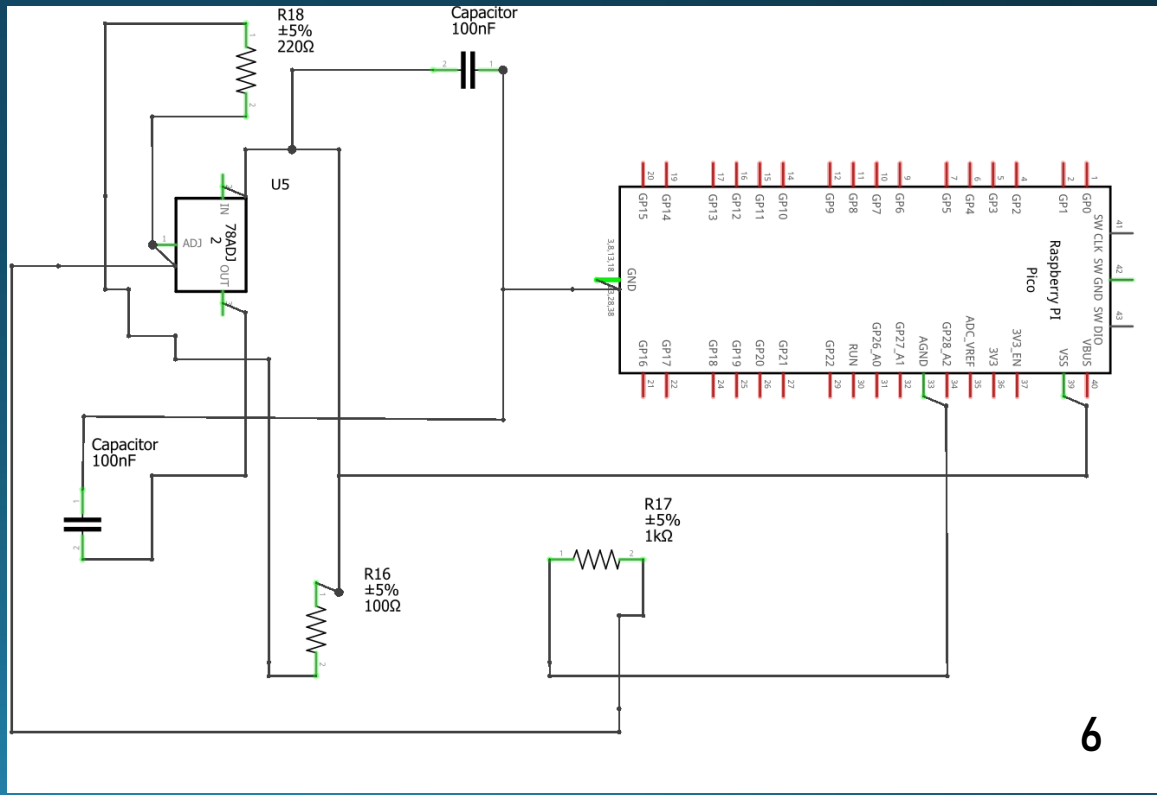




Power Supply and Voltage

- 12 V to 5 V (capacitors and LM317)





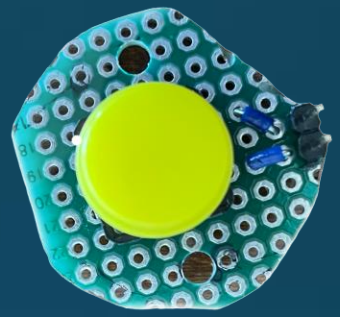
Start Button and LCD

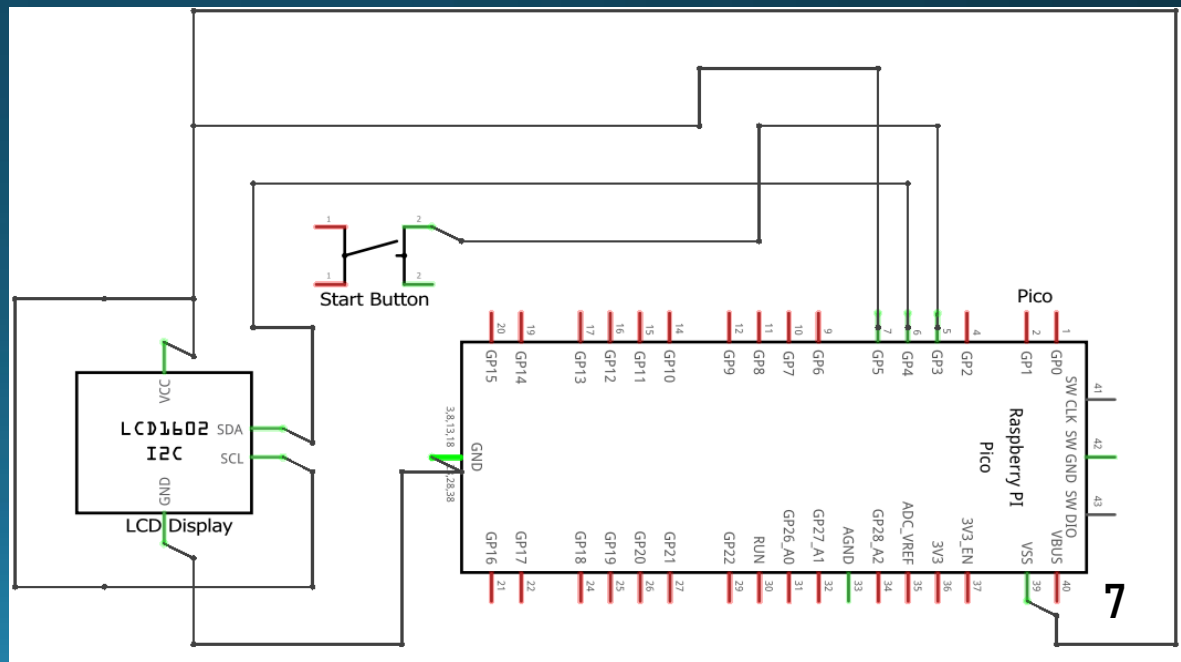
- Simulated EEPROM

```
#include <EEPROM.h>
```

```
void setup(){

  EEPROM.begin(512);
  max_score_ever_int = EEPROM.read(address0);
  max_score_ever_dp = EEPROM.read(address1);
  max_score_ever = (max_score_ever_int*100 + max_score_ever_dp)/100;
```

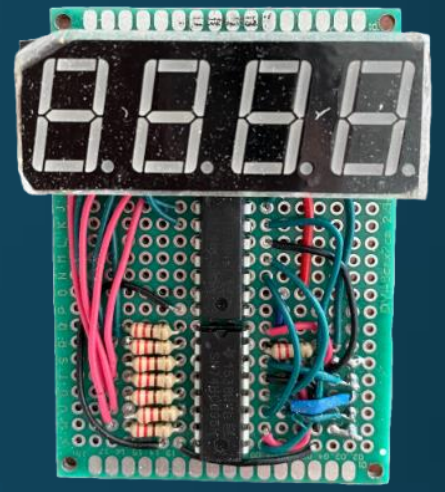
```
EEPROM.write(address0, max_score_ever_int);
EEPROM.commit();
EEPROM.write(address1, max_score_ever_dp);
EEPROM.commit();
```





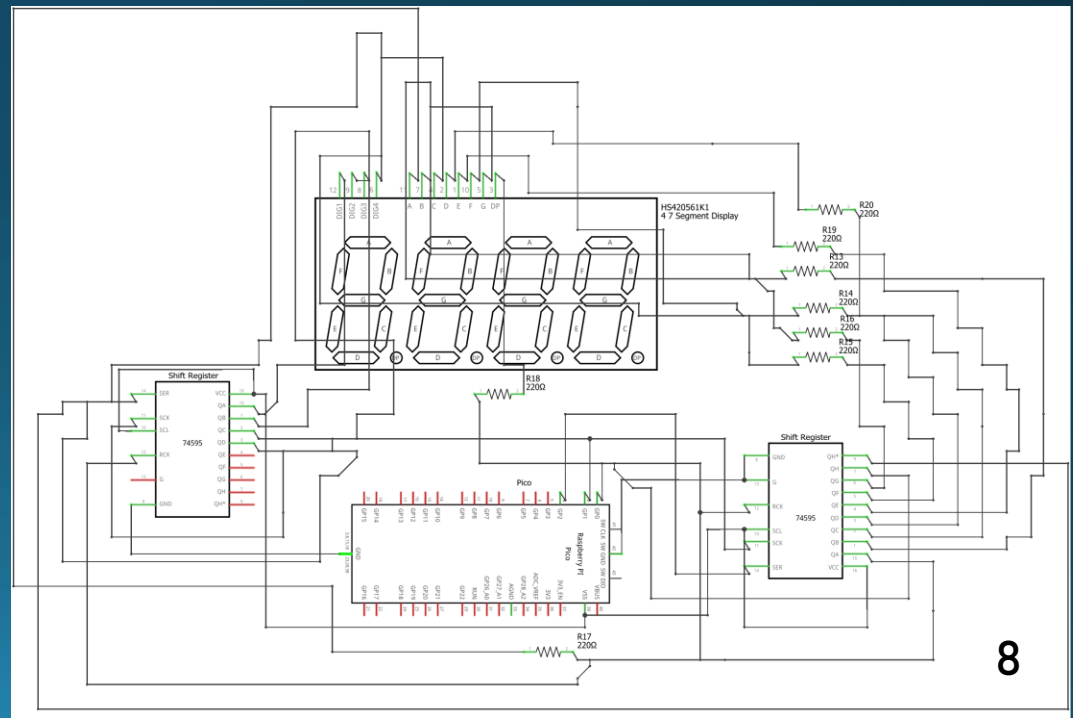
4 7 Segment and Shift Registers

- Multicore processing
- Communication between cores



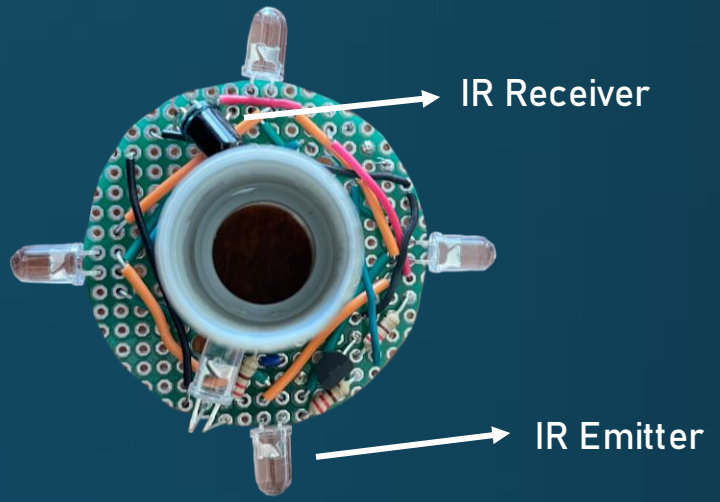
```
void setup1() {
    pinMode(latchPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(startbuttonPin, INPUT);
}

void loop1() {
  if (rp2040.fifo.available()!=0){
    number_to_display_int = rp2040.fifo.pop();
    number_to_display_dp = rp2040.fifo.pop();
  }
  number_to_display = number_to_display_int * 10 + number_to_display_dp;
  writenumber(number_to_display);
}
```

```
rp2040.fifo.push(0);
rp2040.fifo.push(0);
```



RPM Sensor

- Compute rotations per minute
- Handle Interrupt
- Timer Interrupt

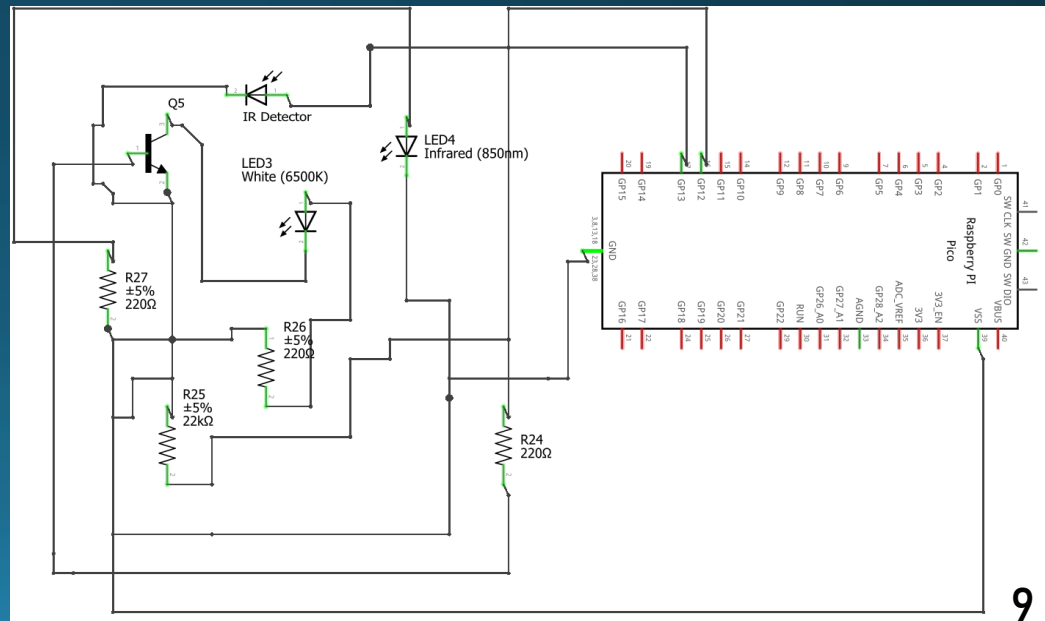



```
#include "RPi_Pico_TimerInterrupt.h"
```

```
attachInterrupt(digitalPinToInterrupt(rpmPIN), interruptIR, CHANGE);
```

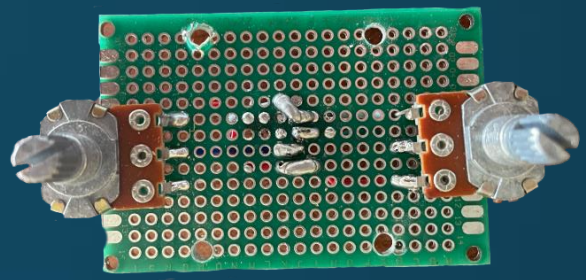
```
IRTimer.attachInterruptInterval(500000, IRTimerHandler);
IRTimer.stopTimer();
```

```
// RPM calculator
void interruptIR() {
  interruptState = digitalRead(rpmPIN);

  if (interruptState && curr_rpm > 30) {
    digitalWrite(ledrpmPin,HIGH);
    IRTimer.restartTimer();
    collisions ++;
  }
  else if (interruptState) turnoffIR();
  else {
    time_made_previous = time_made_now;
    time_made_now = millis();
    count++;
  }
}
```



Coil Button and Potentiometer

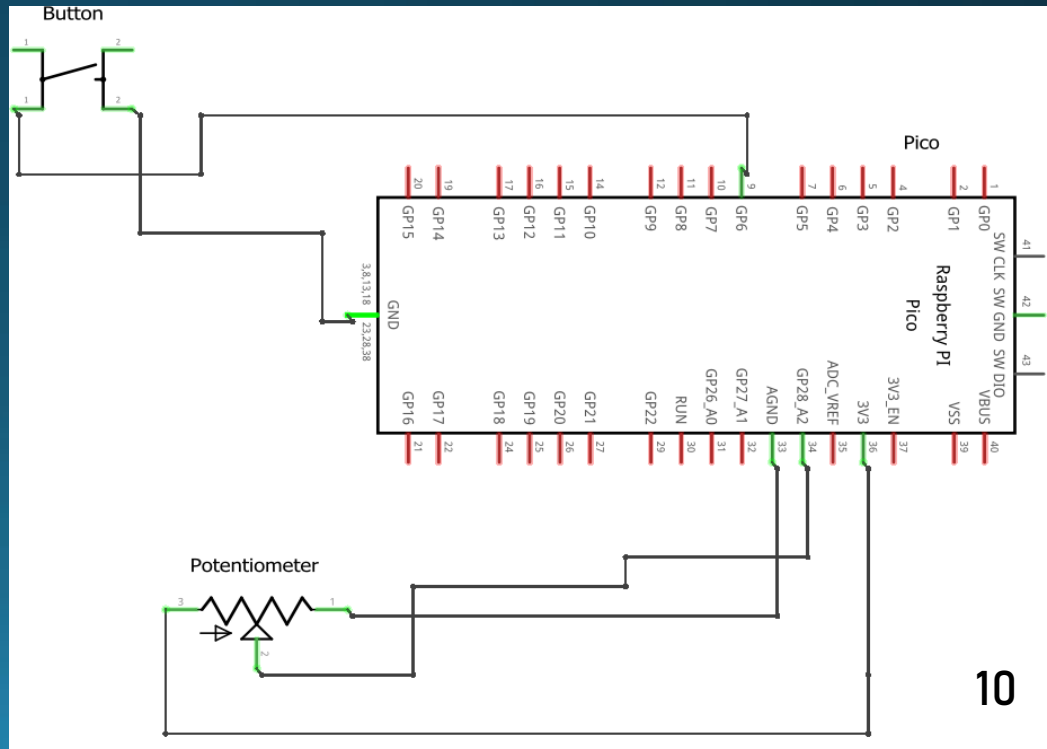
- Handle Interrupt
- Timer Interrupt



```
attachInterrupt(digitalPinToInterrupt(pressbuttonPin), buttonchange, CHANGE);
```

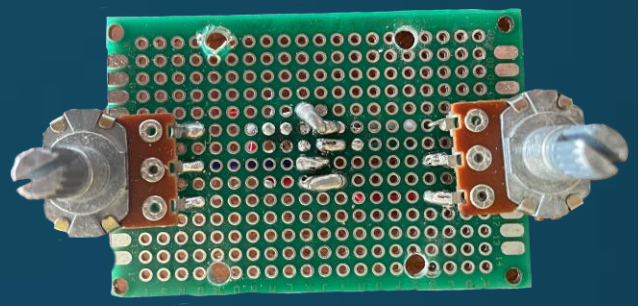
```
ButtonTimer.attachInterruptInterval(500000, ButtonTimerHandler);
ButtonTimer.stopTimer();
```

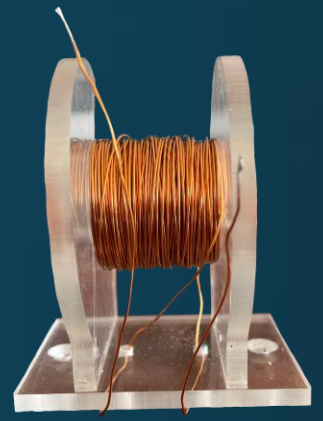
```
void pressButton() {
    pot_coil_Val = analogRead(potcoilPin);

    buttonstatus = false;
    Serial.println(millis() - last_button_change);
    max_press_time = (analogRead(pottimePin)*225 + 24325)/(509*3);
    ButtonTimer.setInterval(max_press_time*1000, ButtonTimerHandler);
    ButtonTimer.restartTimer();
    prev_time_pressed = millis();

    analogWrite(ledpressedPin, pot_coil_Val);
    if (pot_coil_Val > 900) analogWrite(coilPin, max_coil*255);
    else analogWrite(coilPin, (max_coil*255-40)/900 * pot_coil_Val + 40);
}
```



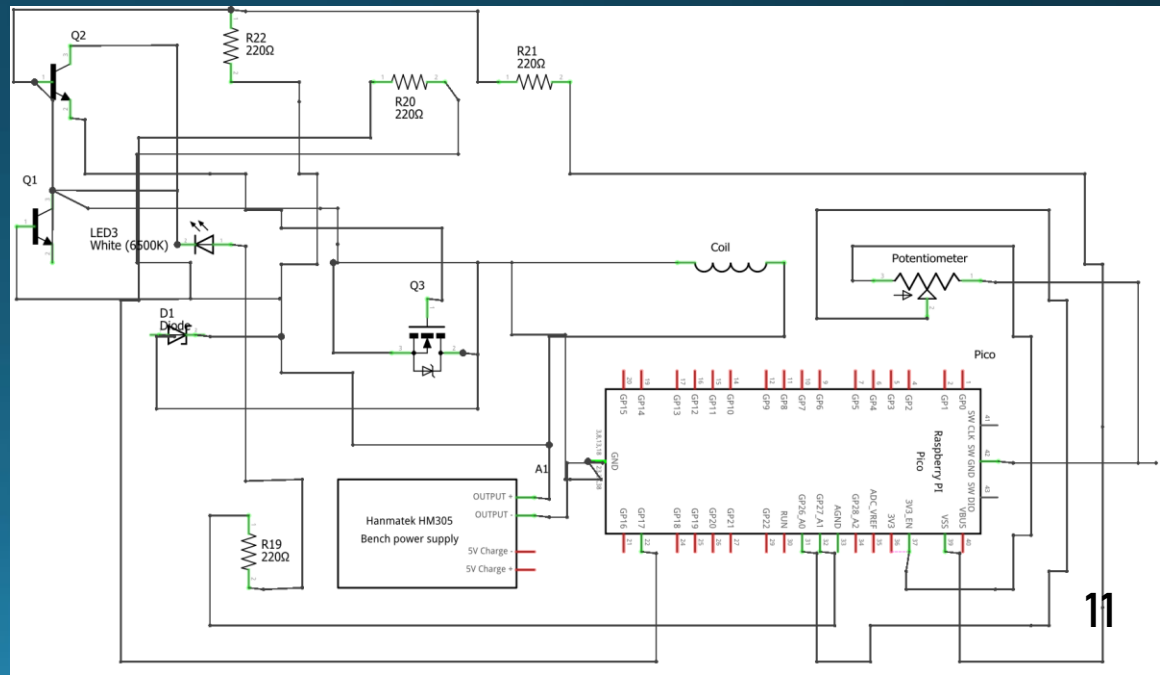
Coil Voltage Regulation

- Potentiometers with 3.3 V





```
void pressButton() {
    pot_coil_Val = analogRead(potcoilPin);

    buttonstatus = false;
    Serial.println(millis() - last_button_change);
    max_press_time = (analogRead(pottimePin)*225 + 24325)/(509*3);
    ButtonTimer.setInterval(max_press_time*1000, ButtonTimerHandler);
    ButtonTimer.restartTimer();
    prev_time_pressed = millis();

    analogWrite(ledpressedPin, pot_coil_Val);
    if (pot_coil_Val > 900) analogWrite(coilPin, max_coil*255);
    else analogWrite(coilPin, (max_coil*255-40)/900 * pot_coil_Val + 40);
}
```



Extra Coding Details

```
void loop(){

  startbuttonState = digitalRead(startbuttonPin);

  if (!startbuttonState) {···
  }

  if (starting){···
  }

  if (running){···
  }

  if (ended){···
  }

  if (idle){···
  }
}
```

- Five Core States
  - Starting
  - Running
  - Ended
  - Idle
  - New High Score
- Efficient State Transitions

```
// States of Game
bool starting = false;
bool running = false;
bool ended = false;
bool idle = true;
bool new_highest_score = false;
```

Final Product

Future Delelopments

- Generic build, for up to 4 players
- Semi-automatic version, with 2 separate tubes
- Calibrate spheres to the same velocity

References

- The Large Hadron Collider | CERN (home.cern)
- Arduino Uno x Raspberry Pi Pico: Qual placa é melhor? – MakerHero
- Nondestructive Evaluation Physics : Magnetism (nde-ed.org)
- Arduino Pico Documentations
- Electromagnetism - How does an Electromagnetic Accelerator work? - Physics Stack Exchange