Microcontrollers and Robotic

Week 3: Interfaces

Fenerbahçe University

Professor & TAs

Prof: Dr. Vecdi Emre Levent

Office: 311

Email: emre.levent@fbu.edu.tr

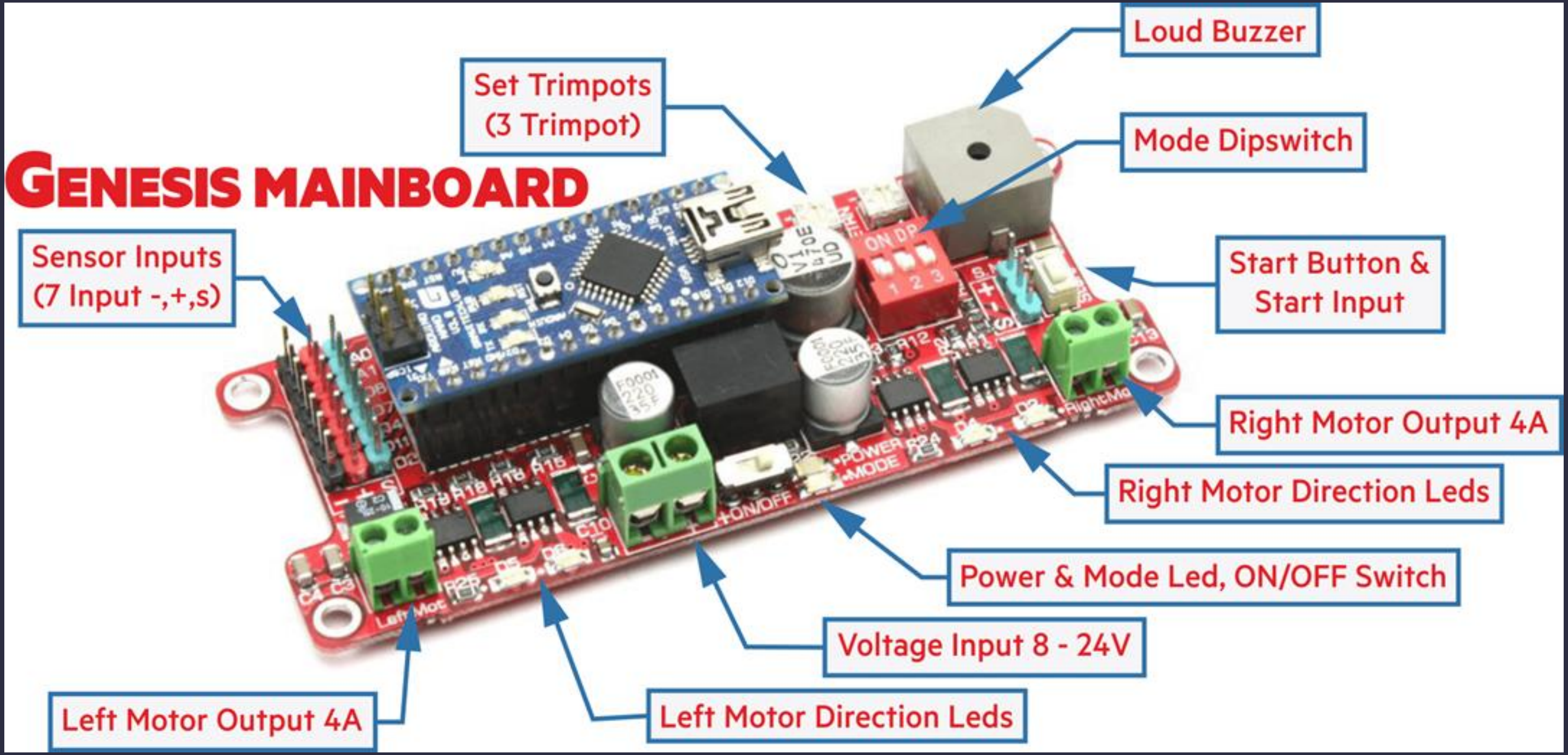
TA: Arş. Gör. Uğur Özbalkan

Office: 311

Email: ugur.ozbalkan@fbu.edu.tr

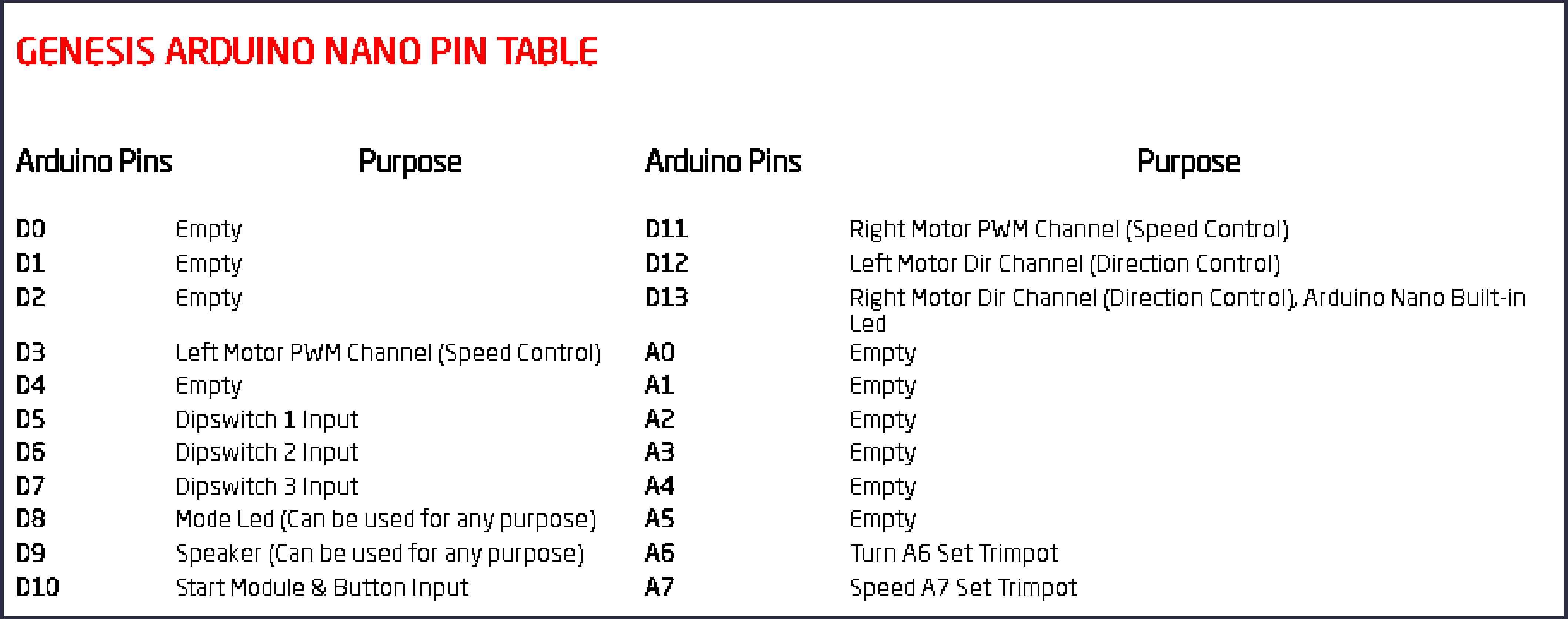
Interfaces

- Arduino Mini Sumo Robot Kit


Loud Buzzer
Set Trimpots (3 Trimpot)
Mode Dipswitch
GENESIS MAINBOARD
Sensor Inputs (7 Input -,+,s)
Start Button & Start Input
Right Motor Output 4A
Right Motor Direction Leds
Power & Mode Led, ON/OFF Switch
Voltage Input 8 - 24V
Left Motor Output 4A
Left Motor Direction Leds

Interfaces

- Arduino Mini Sumo Robot Kit

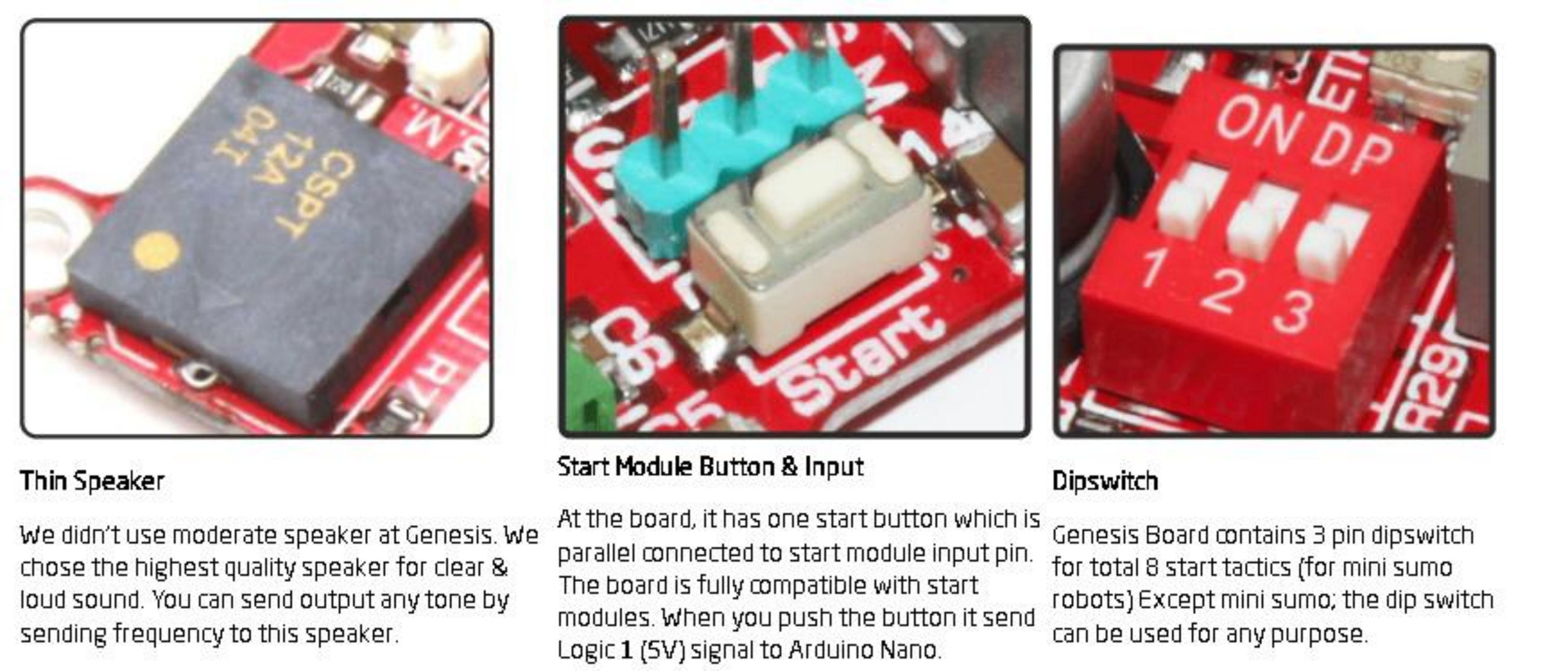


GENESIS ARDUINO NANO PIN TABLE

Arduino Pins	Purpose	Arduino Pins	Purpose
D0	Empty	D11	Right Motor PWM Channel (Speed Control)
D1	Empty	D12	Left Motor Dir Channel (Direction Control)
D2	Empty	D13	Right Motor Dir Channel (Direction Control), Arduino Nano Built-in Led
D3	Left Motor PWM Channel (Speed Control)	A0	Empty
D4	Empty	A1	Empty
D5	Dipswitch 1 Input	A2	Empty
D6	Dipswitch 2 Input	A3	Empty
D7	Dipswitch 3 Input	A4	Empty
D8	Mode Led (Can be used for any purpose)	A5	Empty
D9	Speaker (Can be used for any purpose)	A6	Turn A6 Set Trimpot
D10	Start Module & Button Input	A7	Speed A7 Set Trimpot

Interfaces

- Arduino Mini Sumo Robot Kit



Thin Speaker

We didn't use moderate speaker at Genesis. We chose the highest quality speaker for clear & loud sound. You can send output any tone by sending frequency to this speaker.

Start Module Button & Input

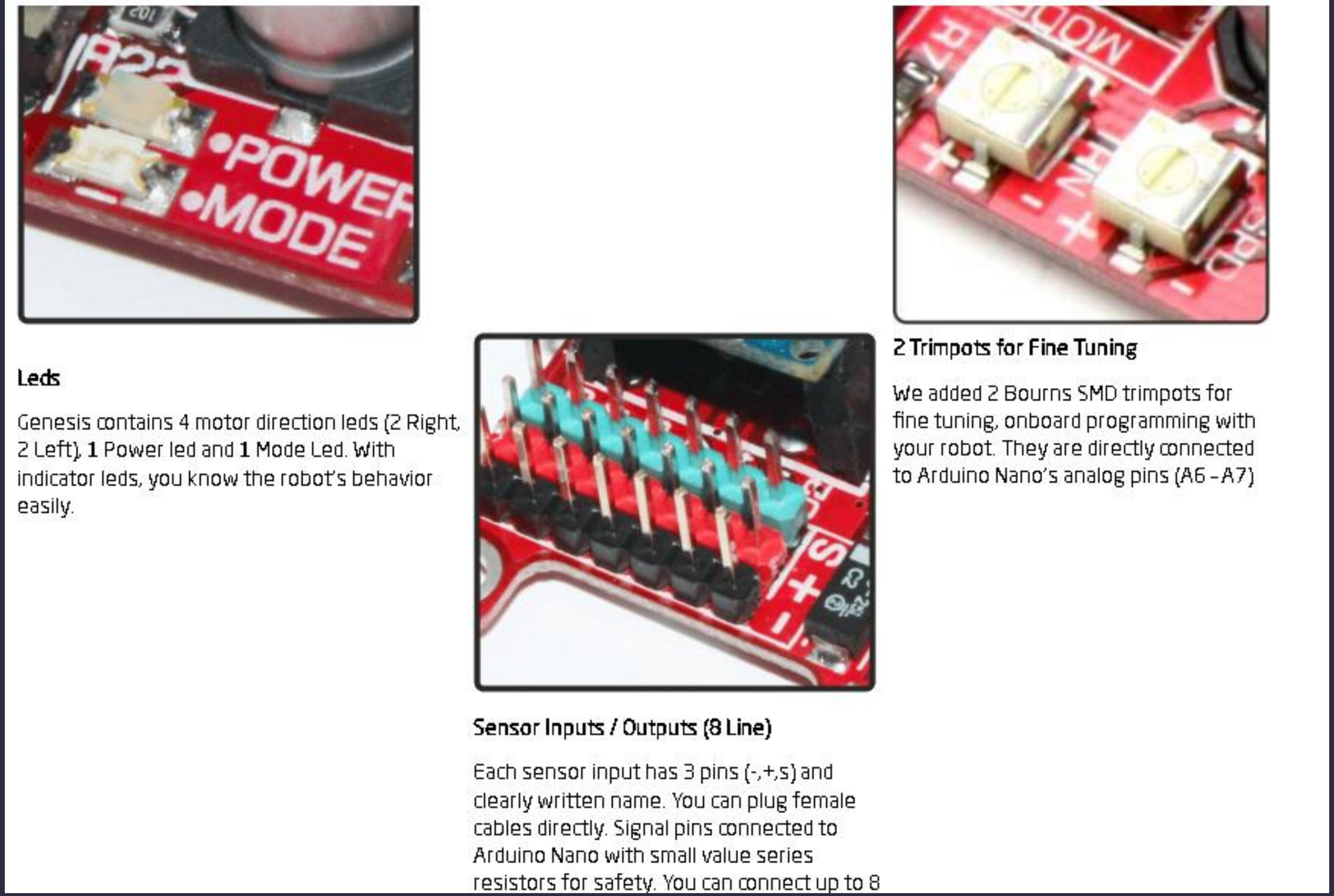
At the board, it has one start button which is parallel connected to start module input pin. The board is fully compatible with start modules. When you push the button it send Logic **1** (5V) signal to Arduino Nano.

Dipswitch

Genesis Board contains 3 pin dipswitch for total 8 start tactics (for mini sumo robots) Except mini sumo; the dip switch can be used for any purpose.

Interfaces

- Arduino Mini Sumo Robot Kit



Leds

Genesis contains 4 motor direction leds (2 Right, 2 Left), **1** Power led and **1** Mode Led. With indicator leds, you know the robot's behavior easily.

Sensor Inputs / Outputs (8 Line)

Each sensor input has 3 pins (-,+,s) and clearly written name. You can plug female cables directly. Signal pins connected to Arduino Nano with small value series resistors for safety. You can connect up to 8

2 Trimpots for Fine Tuning

We added 2 Bourns SMD trimpots for fine tuning, onboard programming with your robot. They are directly connected to Arduino Nano's analog pins (A6 - A7)

Interfaces

- analogRead( ) function

Arduino is able to detect whether there is a voltage applied to one of its pins and capture it through the digitalRead() function.

There is a difference between an on/off sensor (which detects the presence of an object) and an analog sensor, whose value continuously changes.

In order to read this type of sensor, we need a different type of pin.

Interfaces

- analogRead( ) function

In the lower-right part of the Arduino board, you will see six pins marked “Analog In”.

These special pins not only tell whether there is a voltage applied to them, but also its value.

By using the analogRead() function, we can read the voltage applied to one of the pins.

Interfaces

- analogRead( ) function

This function returns a number between 0 and 1023, which represents voltages between 0 and 5 volts.

For example, if there is a voltage of 2.5 V applied to pin number 0, analogRead(0) returns 512.

- analogRead() function Syntax

```
analogRead(pin);
```

Interfaces

- analogRead( ) function

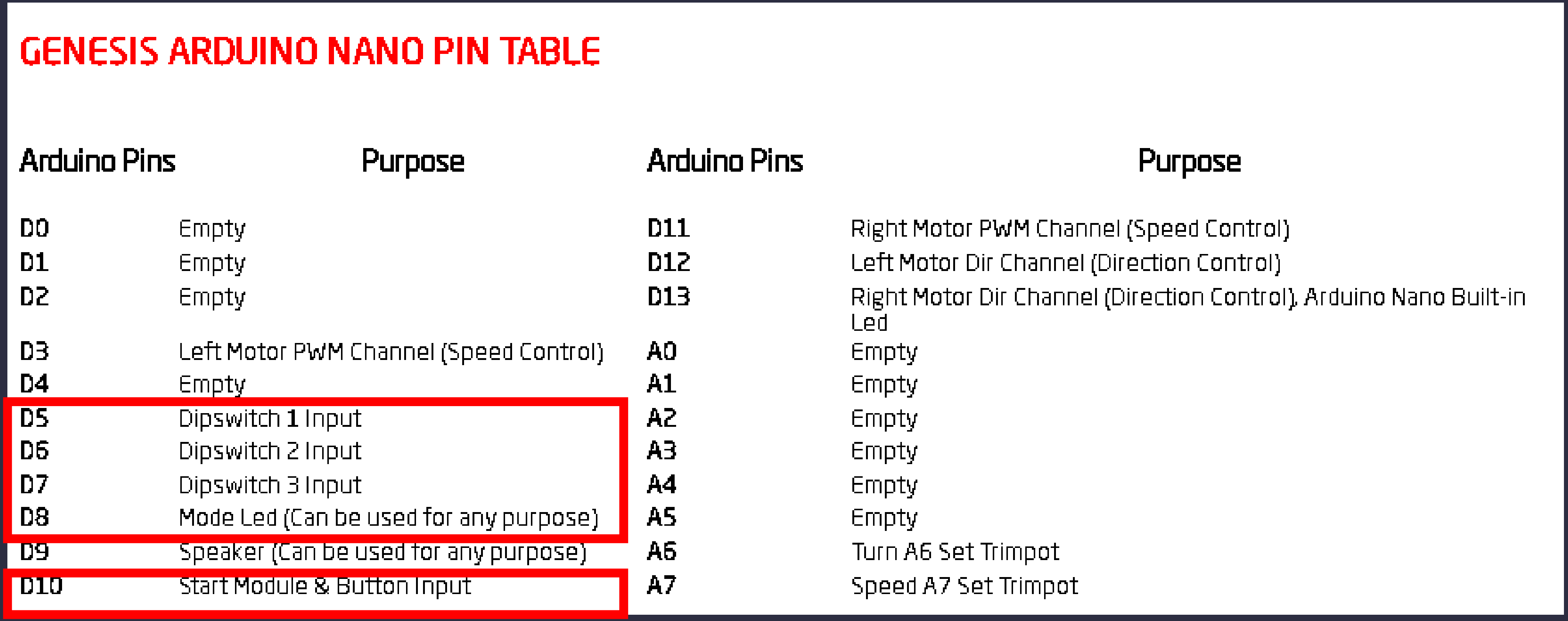
```
int analogPin = 3;//potentiometer wiper (middle terminal)
  // connected to analog pin 3
int val = 0; // variable to store the value read

void setup() {
  Serial.begin(9600); // setup serial
}

void loop() {
  val = analogRead(analogPin); // read the input pin
  Serial.println(val); // debug value
}
```

Interfaces

- GPIOs



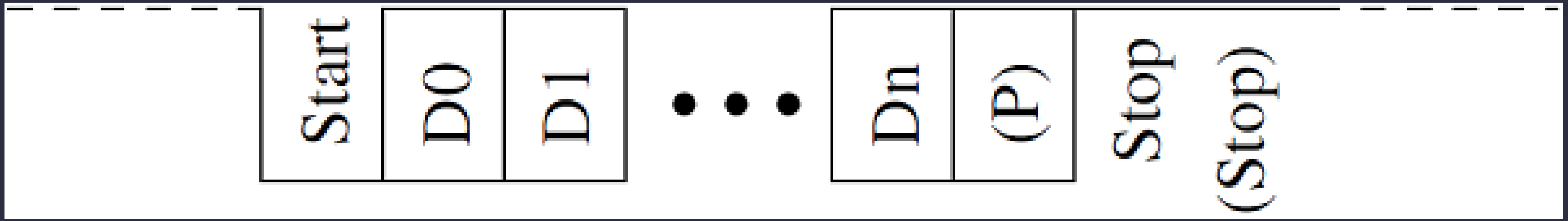
GENESIS ARDUINO NANO PIN TABLE

Arduino Pins	Purpose	Arduino Pins	Purpose
D0	Empty	D11	Right Motor PWM Channel (Speed Control)
D1	Empty	D12	Left Motor Dir Channel (Direction Control)
D2	Empty	D13	Right Motor Dir Channel (Direction Control), Arduino Nano Built-in Led
D3	Left Motor PWM Channel (Speed Control)	A0	Empty
D4	Empty	A1	Empty
D5	Dipswitch 1 Input	A2	Empty
D6	Dipswitch 2 Input	A3	Empty
D7	Dipswitch 3 Input	A4	Empty
D8	Mode Led (Can be used for any purpose)	A5	Empty
D9	Speaker (Can be used for any purpose)	A6	Turn A6 Set Trimpot
D10	Start Module & Button Input	A7	Speed A7 Set Trimpot

Interfaces

- UART

UART (Universal Asynchronous Receiver Transmitter)



Interfaces

UART (Universal Asynchronous Receiver Transmitter)

```
void setup() {
  Serial.begin(9600); //set up serial library baud rate to 9600
  Serial.println("hello world"); //print hello world
}

void loop() {

}
```

Interfaces

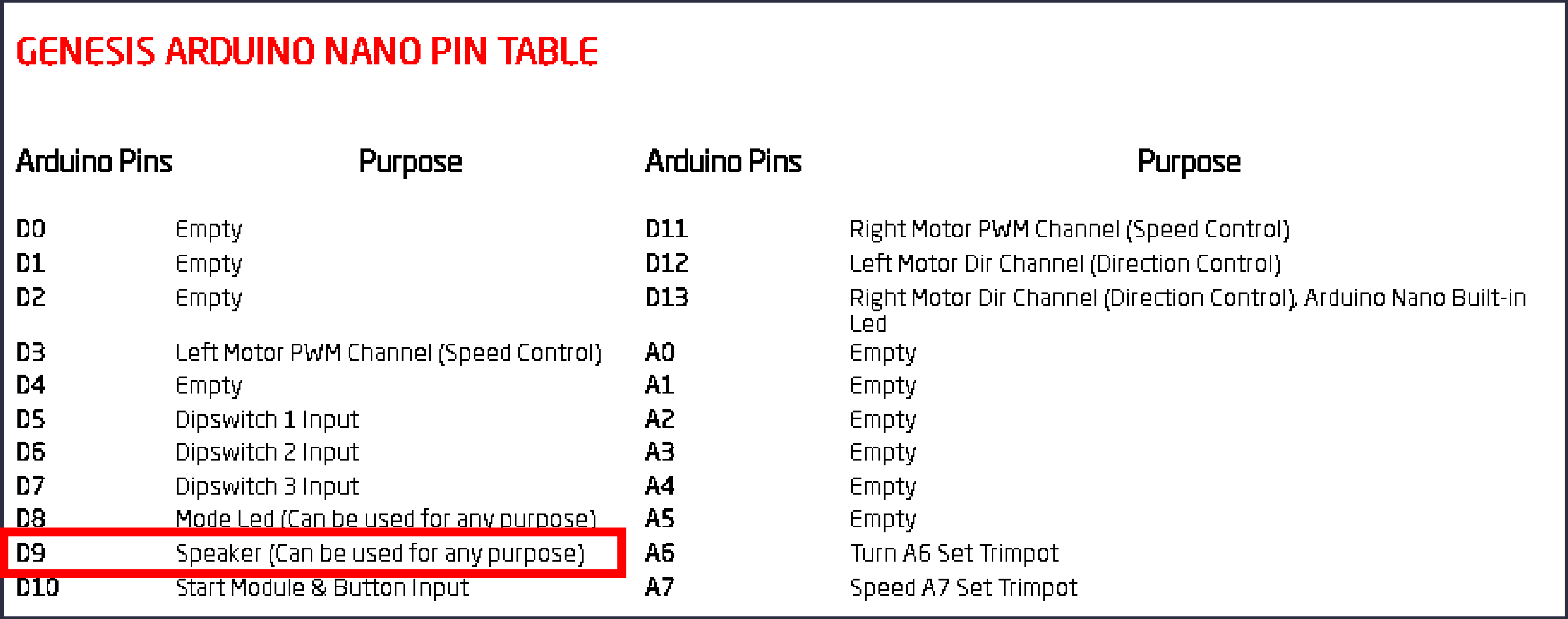
UART (Universal Asynchronous Receiver Transmitter)

```
void setup() {
  Serial.begin(9600); //set up serial library baud rate to 9600
}

void loop() {
  if(Serial.available()){
    Serial.print("I received:"); //print I received
    Serial.write(Serial.read()); //send what you read
  }
}
```

Interfaces

- Buzzer



GENESIS ARDUINO NANO PIN TABLE

Arduino Pins	Purpose	Arduino Pins	Purpose
D0	Empty	D11	Right Motor PWM Channel (Speed Control)
D1	Empty	D12	Left Motor Dir Channel (Direction Control)
D2	Empty	D13	Right Motor Dir Channel (Direction Control), Arduino Nano Built-in Led
D3	Left Motor PWM Channel (Speed Control)	A0	Empty
D4	Empty	A1	Empty
D5	Dipswitch 1 Input	A2	Empty
D6	Dipswitch 2 Input	A3	Empty
D7	Dipswitch 3 Input	A4	Empty
D8	Mode Led (Can be used for any purpose)	A5	Empty
D9	Speaker (Can be used for any purpose)	A6	Turn A6 Set Trimpot
D10	Start Module & Button Input	A7	Speed A7 Set Trimpot

Interfaces

- Buzzer

tone(): Generates a square wave of the specified frequency (and 50% duty cycle) on a pin. A duration can be specified, otherwise the wave continues until a call to noTone(). The pin can be connected to a piezo buzzer or other speaker to play tones.

tone(pin, frequency)

noTone(): Stops the generation of a square wave triggered by tone(). Has no effect if no tone is being generated.

noTone(pin)

Interfaces

- Buzzer

```
int buzzerPin = 9;
int notaSayisi = 8;
int C = 262;
int D = 294;
int E = 330;
int F = 349;
int G = 392;
int A = 440;
int B = 494;
int C_ = 523;
int notalar[] = {C, D, E, F, G, A, B, C_};
```

```
void setup()
{
  for (int i = 0; i < notaSayisi; i++)
  {
    tone(buzzerPin, notalar[i]);
    delay(500);
    noTone(buzzerPin);
    delay(20);
  }
  noTone(buzzerPin);
}

void loop()
{
}
```

Interfaces

- Buzzer

```
#include "pitches.h"

int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3,
NOTE_G3, 0, NOTE_B3, NOTE_C4
};

int noteDurations[] = {

  4, 8, 8, 4, 4, 4, 4, 4
};
```

```
void setup() {

  for (int thisNote = 0; thisNote < 8; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);

  }
}

void loop() {

  // no need to repeat the melody.
}
```