AUTOMATIC GUITAR PLAYER

Jordan JACOBI

Gaëthan LEGRAND

Arnaud PEETERS

Benjamin VANDENBUSSCHE

Nour VERHEGGEN

2019

4MEI-EO ECAM

INTRODUCTION

To be written at the end of the work

OBJECTIVES

- Choose the appropriate technology to control the robot
    - Controllers
    - Actuators
    - Electronics
    - Power supply
- Design the mechanical structure of the project
    - Mount of the guitar
    - Rag & Pinion system
- Design the electronics for the project
- Translate the ABC source file into a usable input
- Control the actuators to play the full song

TIMELINE

	Jordan	Arnaud	Gaëthan	Nour	Benjamin
Week 1	Test of the Stepper motor and its characteristics				
Week 2	Reflexion about the power supply of the whole robot				
Week 3	I2C Communication between Arduino & Raspberry Pi				
Week 4	First part of the design of the PCB				
Week 5	Design of a component : Mount + Guide for the rails (free wheels part)				
Week 6	Measurements of the structure and support and installation				
Week 7	Fixed numerous bugs in the main python program & set gpio (boot time & config)				

Week 8	GUI & scripts				

COMPONENTS

STEP MOTORS

The step motors have been chosen for their ability to provide a distance without the need of other sensors. Because they are controlled using pulses - making them move the same distance, a step - we always know the position of the motor, assuming there are no missed steps.

The motors used are the NEMA17HS4401 composed of 2 pairs of coils :

- A+ : Red
- A- : Green
- B+ : Yellow
- B- : Blue

The driver A4988 controls these motors.

DRIVER A4988

The driver controls the step motors using inputs from a Raspberry Pi. The minimal wiring diagram is shown below fig. X.

The inputs are :

- DIR : Determines the direction of the motor, allowing to go both directions.
- STEP : Each rising edge on this pin make the motor move 1 step - **full rotation = 200 steps.**
- /SLEEP : When given a low logic voltage, put the driver into sleep mode for lower energy consumption. **Needs 1ms to go back to run mode**.
- /RESET : Resets the driver.

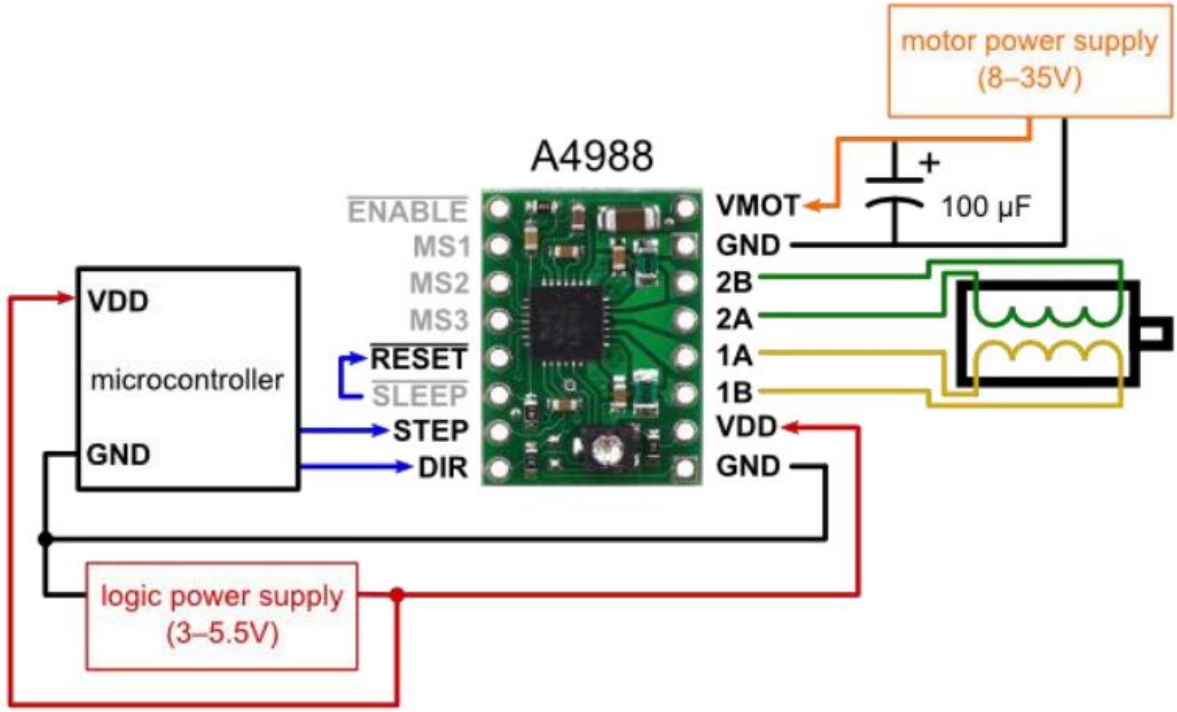
Not used inputs :

- MSX : Microstep inputs. Allow use of moves of smaller steps (to 1/16 of a step).
- /ENABLE : Disables the outputs of the driver.

We can adjust the max output current on the driver's screw.

Initial tests with an Arduino code revealed we can only operate this driver **using command signals on the STEP pin from 40Hz to 1000Hz.** Further tests with the rag and pinion and the real Python code showed some missed steps at the highests frequencies.

Therefore, the final operating frequency is around **600Hz.**



SERVOMOTORS

IR SENSORS

POWER SUPPLY

The whole project is not meant to be moved regularly. Thus it will be powered from a static power supply like the electrical network (230V AC - 50 Hz) or a DC power supply.

	AVERAGE CURRENT	MAX CURRENT	VOLTAGE
STEP MOTOR + DRIVER (POWER FOR ONE UNIT)	IDLE : 0.17A NORMAL : 0.2A	WITH LOAD : 0.35A	12V
ARDUINO (WITH SERVOS)	50mA	+100mA	12V

RASPBERRY PI	200mA	500mA	5V

Those values are only indicative because the tests have been done with arbitrary codes and constraints applied to the motors.

Due to the relative weakness of the Raspberry Pi to disturbances if supplied by its pin Vin, it will be powered from the network using a regular power supply. It will then provide the logical voltage to the rest of the project (I2C bus, drivers, ...) A power supply module (or if not available a 12V DC power supply) will provide the Arduino supply and the motors power.

MECHANICAL PARTS

RAG & PINION

MOUNT + GUIDE

SENSOR MOUNT

PCB

CONTROL LOGIC

I2C

Why I2C ? Serial communication, the other possibility, relies on the bitrate whereas I2C is synchronous. Furthermore, it is a master/slave protocol allowing an easy communication between the controllers.

The communication between the Arduino and the Raspberry Pi is used to control the right hand of the robot i.e. the 6 servomotors. The bus is implemented on the PCB using 10kohm resistors and the addresses are fixed in the software.

Tools required to use I2C :

- “Wire.h” on the Arduino
- SMBUS on the Raspberry Pi
- Devtools on the Raspberry Pi

apt-get install i2c-tools python-smbus python3-smbus

Do not forget to enable I2C in the configuration of the Raspberry Pi :

raspi-config

then select “Interfacing options” => “I2C” => “Yes”

	Arduino	Raspberry Pi (bus 1)
SDA	20	3 board
SCL	21	5 board

ABC TRANSLATION

NODE-RED GUI

Install node-red + node-red-dashboard via graphical interface on node-red interface

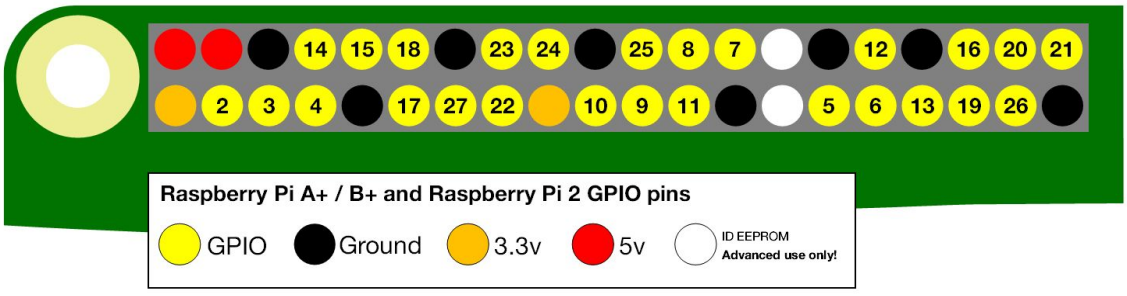
access through **agp:1880/ui**

Scripts & logic

MAIN PROGRAM

sudo apt-get install rpi.gpio

GPIO configuration



GPIO can be set as inputs or outputs at boot time using the file **/boot/config.txt**

*GPIO are currently defined in the main python script. This part should be moved to a

configuration file.

	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6
DIR	27	9	6	26	15	25
STEP	4	22	11	13	14	23
/SLEEP	17	10	5	19	18	24
IR	8	7	12	20	16	21

INTERFACE

The program on the Raspberry Pi will be controlled by a graphical interface.

? choices **node Red** ? Angular.js ? backend Django ?

pip install Django

CONCLUSION

To be written at the end of the work

EXTERNAL LINKS

1. Driver datasheets : http://www.communica.co.za/Content/Catalog/Documents/D0050510630.pdf
   https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf
2. Stepper motor datasheet :
   https://laborjag.com/venta/3d-printer-cnc/nema-17-stepper-motor-17hs4401/
   https://datasheetspdf.com/pdf-file/928661/MotionKing/17HS4401/1
3. Arduino tutorial to control a stepper motor :
   https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/
4. Arduino note about its power supply : https://shop.evilmadscientist.com/productsmenu/564
5. Buck datasheets :
   https://www.banggood.com/XH-M401-DC-DC-Step-Down-Module-XL4016E1-High-Power-DC-Voltage-Regulator-Maximum-8A-p-1314177.html?akmClientCountry=BE&currency=AUD&createTmp=1&utm_source=commissionfactory&utm_medium=aff&utm_content=31940&cfclick=51b75fec7480499480a70288e877e79b&cur_warehouse=CN http://www.xlsemi.com/datasheet/xl4016%20datasheet.pdf
6. I2C communication between Arduino and Raspberry Pi :
   https://www.pihomeserver.fr/2013/08/13/raspberry-pi-home-server-arduino-lier-les-deux-via-bus-i2c/

APPENDICES

Add useful and frequently used vocabulary to this table

English	French
beat	temps (musical)
fretboard	touche (de la guitare)
neck (of the guitar)	manche (de la guitare)
rag and pinion	crémaillère (+ roue)
free wheel	roue folle
drive belt	courroie de tranmission

English	French
A	La
B	Si
C	Do
D	Ré
E	Mi
F	Fa
G	Sol