Embedded Systems
Advanced Operating Systems Project

Whistle Robot

Version 0.0
??/?/2018

Borgo Daniele	894110
Fusca Yuri	??????
Indirli Fabrizio	899892

Contents

1 Introduction **1**
1.1 Purpose 1
1.2 Description of the robot 1
1.3 Game logic 1
1.4 Code structure 1

2 Controller **2**

3 Microphone **3**
3.1 The MP45DT02 microphone 3
3.2 Overview on sound acquisition and processing 4
3.3 Microphone initialization 5
3.4 Sound acquisition and processing 5

4 Wireless Communication **6**

5 Motors **7**

6 Bibliography **8**

1 Introduction

1.1 Purpose

The goal of this project is to design a game for 2 players, where the STM32s microphone is used to control a robot, and the player who can defeat the opponent robot wins.
This project is a variant of the project proposal where the STM32F4 was used in a game for 2 or more players to choose and recognise a frequency with the microphone.
In our project, we use the embedded microphone of the board to controls the stepper motors and move the robot on the surrounding environment.

1.2 Description of the robot

The robot will have a cylindrical shape and it will be equipped with wheels on the bottom that will allow it to move forward, backward and rotate on itself.
The engines that will allow the wheels to rotate are directly controlled by the STM32F4, the brain of our robot. We will use 3D printer to build the body of the robot.

1.3 Game logic

The game consists of a competition between 2 or more players. The players will control their robot through sounds, with the aim of attacking and overturning the opposing robot.
A specific movement of the robot will correspond to each sound (frequency):

- Forward movement
- Backward movement
- Rotation on itself

The rotation on itself is useful to turn around and attack the opposing robot

1.4 Code structure

We decided to structure the code in many components, in order to make the project extendable, easy to edit and easy to readapt.
The main core of the robot is represented by the controller. All other components of the robot will be connected to the controller.
The frequency analyser will convert the analog data obtained with the microphone into digital numbers.
The controller will receive the information from the frequency analyser and it will use this information to directly control the wheels of the robot.

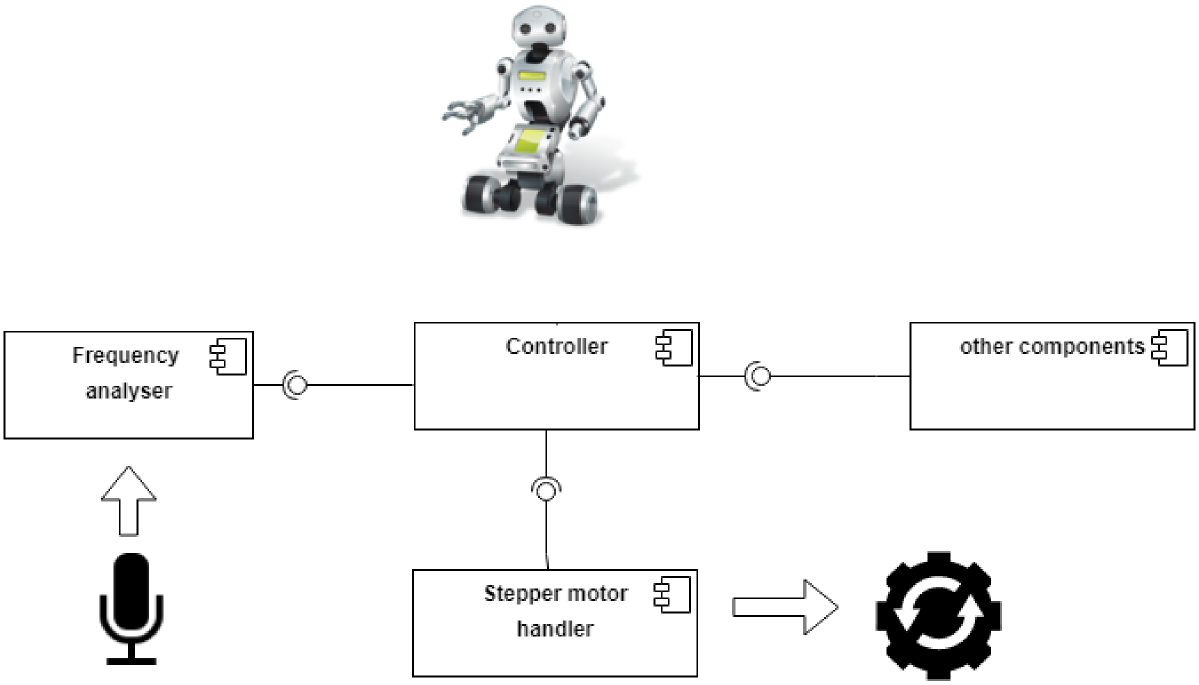

Figure 1: This is a general overview of the system.

2 Controller

3 Microphone

The STM32D407VG-Discovery board is equipped with a MP45DT02 MEMS microphone and a CS43L22 DAC for audio acquisition and processing. In this project, we will use these devices to detect the sounds' frequency, which will determine the commands given to the robot.

3.1 The MP45DT02 microphone

The MP45DT02-M is a compact, low-power, topport, omnidirectional, digital MEMS microphone. It's soldered on the top of the STM32F407G-DISC1 board, in the bottom-right corner.

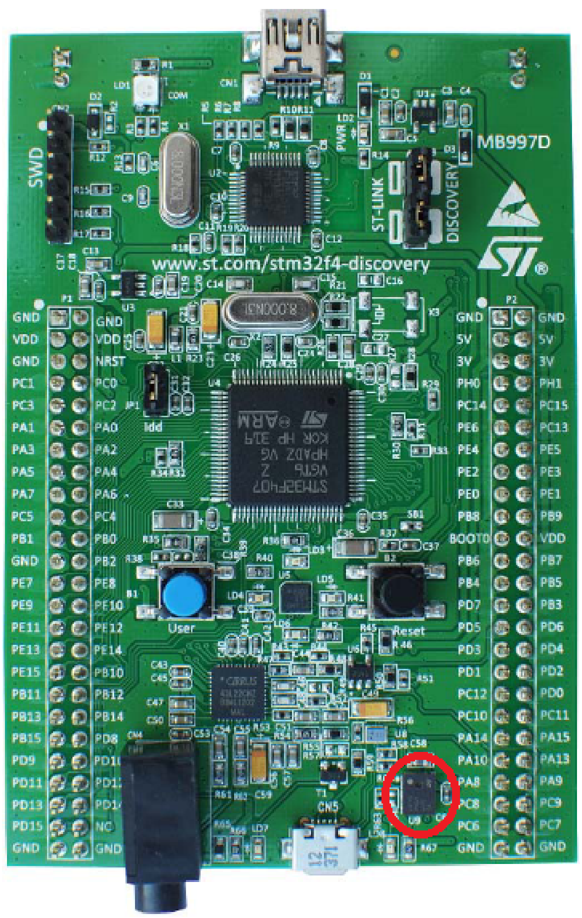

Figure 2: View of the upper surface of the board, with the microphone highlighted in the circle.

In its bottom surface, it has 6 pins:

1. **GND** - Connected to the GND of the board
2. **LR** - Channel selection: used because the microphone is designed to allow stereo audio capture. If it is connected to GND, the MP45DT02 is placed in "left" channel mode: a sample is latched to the data output pin (PDM) on a falling edge of the clock, while on the rising clock edge, the output is set to high impedance. If the LR pin is connected to Vdd then the device operates in "right" channel mode and the MP45DT02 latches it's sample to PDM on a clock rising edge, setting the pin to high impedance on the falling edge of the clock.
3. **GND** - Connected to the GND of the board
4. **CLK** - Synchronization input clock: this pin is connected to the PB10 port of the board. The clock signal determines the sampling frequency.
5. **DOUT** - PDM Data Output. Connected to the PC03 pin of the board's GPIO.
6. **VDD** - Power supply.

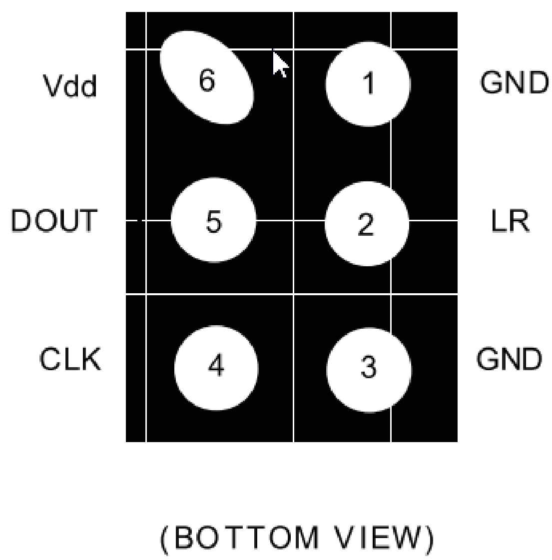

Figure 3: Pins on the bottom of the microphone.

3.2 Overview on sound acquisition and processing

The sound is acquired by the microphone, and the communication with it happens through I2C.

The microphone acquires data in PDM format: each sample is a single bit, so the acquisition is a stream of bits; an high amplitude is represented with an high density of '1' bits in the bitstream.
The sampling frequency of the ADC is 11000 Hz.
In order to process the data, it has to be converted in PCM format: this is done by doing **CIC filtering**, with a decimation factor of 16 (each 16-bits sequence of 1-bit PDM samples is converted in one 16-bit PCM sample).
Then, an FFT analysis is performed on 4096 samples at a time to extract the fundamental frequency and the amplitude of the sound.

3.3 Microphone initialization

The initialization of the sound acquisition system comprehends several steps:

1. SPI, DMA and General Purpose ports B and C are enabled through RCC
2. GPIO ports are configured in *Alternate* mode
3. SPI is set to work in I2S mode
4. interrupt handling for DMA is configured

3.4 Sound acquisition and processing

Each time a new sample is ready:

1. Read 16 bits from SPI and transfer them in RAM through DMA
2. Convert 16 PDM samples in one 16-bit PCM sample via **CIC filtering** with a decimation factor of 16
3. Performs FFT through the *arm_cfft_radix4_f32* module
4. Calculates amplitude of each frequency with the *arm_cmplx_mag_f32* module
5. calculates the Harmonic Product Spectrum to find the fundamental frequency (and its amplitude) in the vector
6. Stores the values of the fundamental frequency and its amplitude in dedicated variables, then calls the **callback function** to react to the detected frequency.

4 Wireless Communication

5 Motors

6 Bibliography