

# Ranger Vervet

A fauna conservation and community protection device



Marco Carpentiero - Armando Cavallaro - Riccardo Califano

Course of  
**EMBEDDED SYSTEMS**

System Documentation and Reference Manual

Academic year 2018/2019

In collaboration with

## Università degli Studi di Salerno

Dipartimento di Ingegneria dell'Informazione ed Elettrica e  
Matematica Applicata



.DIEM



# Contents

<b>Acronyms list</b>	<b>iv</b>
<b>1 Preface</b>	<b>1</b>
1.1 About this manual . . . . .	2
1.2 Manual usage . . . . .	2
1.3 Notation and conventions . . . . .	2
<b>2 Introduction</b>	<b>3</b>
2.1 Backmatter . . . . .	4
<b>3 Requirements</b>	<b>5</b>
3.1 Functional Requirements . . . . .	6
3.2 Non-Functional Requirements . . . . .	8
3.3 Structural Requirements . . . . .	9
<b>4 Hardware Specifications</b>	<b>10</b>
4.1 Introduction . . . . .	11
4.2 STMF401RE Microcontroller Unit . . . . .	12
4.3 SIMCOM 800L GSM Module . . . . .	15
4.3.1 SIMCOM800L GSM Module Hardware - Software Interface . . . . .	18
4.4 OmniVision OV7670 Camera Module . . . . .	18
4.4.1 OV7670 Camera Module Hardware-Software interface . . . . .	21
4.5 KY-037 Microphone Sensor Module . . . . .	22
4.5.1 KY-037 Hardware-Software interface . . . . .	23
4.6 Adafruit NeoPixel FeatherWing LED Matrix . . . . .	23
4.6.1 AdaFruit NeoPixel FeatherWing Hardware-Software interface . . . . .	25

4.7	Audio Speaker . . . . .	26
4.7.1	Audio Speaker Hardware-Software interface . . . . .	26
4.8	28BYJ-48 Stepper Motor . . . . .	27
4.8.1	28BYJ-48 Hardware-Software interface . . . . .	29
4.9	Electrical Connections . . . . .	30
4.9.1	SIMCOM800L - STM32F401RE connection . . . . .	30
4.9.2	OV7670 - STM32F401RE connection . . . . .	31
4.9.3	KY037 - STM32F401RE connection . . . . .	31
4.9.4	Adafruit NeoPixel FeatherWing - STM32F401RE connection . . . . .	32
4.9.5	Audio Speaker - STM32F401RE connection . . . . .	32
4.9.6	28BYJ-48 - STM32F401RE connection . . . . .	33
4.9.7	Ranger Vervet System wiring diagram . . . . .	34
<b>5</b>	<b>Software Specifications</b>	<b>35</b>
5.1	Software Architecture . . . . .	36
5.1.1	Modules Details . . . . .	37
5.2	Software dynamic behavior . . . . .	45
5.3	Ranger Vervet Skills . . . . .	49
5.3.1	Audio Localization . . . . .	49
5.3.2	Recorded audio signal construction . . . . .	50
5.3.3	Audio Classification . . . . .	50
<b>References</b>		<b>53</b>
<b>List of figure</b>		<b>54</b>
<b>List of tables</b>		<b>56</b>

# Acronyms list

**2G** Second-generation cellular technology

**ABF** Automatic Band Filter

**ABLC** Automatic Black Level Calibration

**ADC** Analog to Digital Converter/Conversion

**AEC** Automatic Exposure Control

**AGC** Automatic Gain Control

**AHB** Advanced High-Performance Bus

**AOA** Angle of arrival

**APB** Advanced Peripheral Bus

**ART** Adaptive Real-Time accelerator

**AWB** Automatic White Balance

**CIF** Common Intermediate Format

**CMOS** Complementary Metal-Oxide Semiconductor

**CMSIS** Cortex Microcontroller Software Interface Standard

**CPU** Central Processing Unit

**CRC** Cyclic Redundancy Check

**DCE** Data Communication Equipment

**DMA** Direct Memory Access

**DSP** Digital Signal Processing

**DTE** Data Terminal Equipment

**EGPRS** Enhanced General Packet Radio Service

**FIFO** First In First Out

**FIR** Finite Impulse Response

**FPU** Floating Point Unit

**GPRS** General Packet Radio Service

**HAL** Hardware Abstraction Layer

**HTTP** Hypertext Transfer Protocol

**I/O** Input Output

**I2C** Inter Integrated Circuit

**I2S** Inter-IC Sound

**IC** Input Capture

**IP** Internet Protocol

**ISR** Interrupt Service Routine

**LCD** Liquid Crystal Display

**LED** Light Emitting Diode

**LMS** Least Mean Squares

**MCU** MicroController Unit

**MPU** Memory Protection Unit

**MUSIC** MUltiple Slgnal Classification

**OC** Output Compare

**PHY** Physical Layer

**PLL** Phase-locked loop

**PWM** Pulse Width Modulation

**QCIF** Quarter Common Intermediate Format

**QQVGA** Quarter Quarter Video Graphics Array

**QVGA** Quarter Video Graphics Array

**RGB** Red Green Blue

**RISC** Reduced Instruction Set Computer

**RTC** Real Time Clock

**SCCB** Serial Camera Control Bus -

**SDIO** Secure Digital Input Output

**SIM** Subscriber Identity Module

**SMS** Short Message Service

**SPI** Serial Peripheral Interface

**SRAM** Static Random Access Memory

**STM** STMicroelectronics

**UART** Universal Asynchronous Receiver-Transmitter

**USART** Universal Synchronous Asynchronous Receiver-Transmitter

**TCP** Transmission Control Protocol

**TDOA** Time Difference Of Arrival

**UML** Unified Modeling Language

**USB** Universal serial bus

**USB-OTG** Universal Serial Bus On The Go

**VGA** Video Graphics Array

# **Chapter 1**

## **Preface**

## 1.1 About this manual

The purpose of this manual is to provide for the reader all the necessary information about the Ranger Vervet device, comprehensive of design issues and features. These will be explored in hardware and software terms. It is directed to engineers and software experts but does not have the intention to fully cover all the arguments cited, for which further readings are recommended.

## 1.2 Manual usage

The "Introduction" chapter contains the problem statement that led to the design and implementation of the device.

The "Requirements" chapter sums up the different requirements that have been stated.

The "Hardware Specifications" chapter digs into some of the details of the devices concurring to the whole Ranger Vervet device.

Finally, the "Software Specifications" chapter covers the most interesting and important aspects of the software driving the system.

## 1.3 Notation and conventions

- File names are reported in typewriter font;
- Voltage supply wires in electrical schematics are depicted in red (positive) and black (negative);
- Software information provide a brief description of its function, input parameters, output values.

# **Chapter 2**

## **Introduction**

## 2.1 Backmatter

A quite common monkey species of the South-East of Africa is the Vervet Monkey. This monkey has different predators, both from air and land, such as eagles, hawks, snakes, pythons, leopards, hyenas and more. This funny and very intelligent creature reacts to the different predators with a specific alarm call to warn its community about the danger. A clever idea may be to detect its alarm and process it through a numerical system in order to achieve a purpose. For example, its ability could be used in a sort of anti-intrusion system for protected natural reserves where endangered species are kept and exposed to possible predators attacks. Also poachers and illegal hunters could be spotted by these monkeys. In particular any land predator might be confused by a leopard, increasing the importance of the alarm in order to report any other terrestrial threat. From another point of view, this peculiarity might also be used for safety purposes in some touristic areas, in which Vervet Monkeys live in touch with humans to protect them from other dangerous savage animals belonging to the local fauna.



**Figure 2.1:** Friendly Vervet Monkeys.

The aim of this system is to exploit and recognize the sounds produced by Vervet Monkeys to identify the threat and react accordingly to it, mainly by scaring away the possible attacker. For more precise specifications check chapter 3.

The scientific dissertations behind this idea can be examined in [1] and [2], where the vocal calls of these animals have been finally quantitatively studied from the signal processing point of view, considering how many linguistic and sociological studies developed in the previous decades about Vervet Monkeys communities.

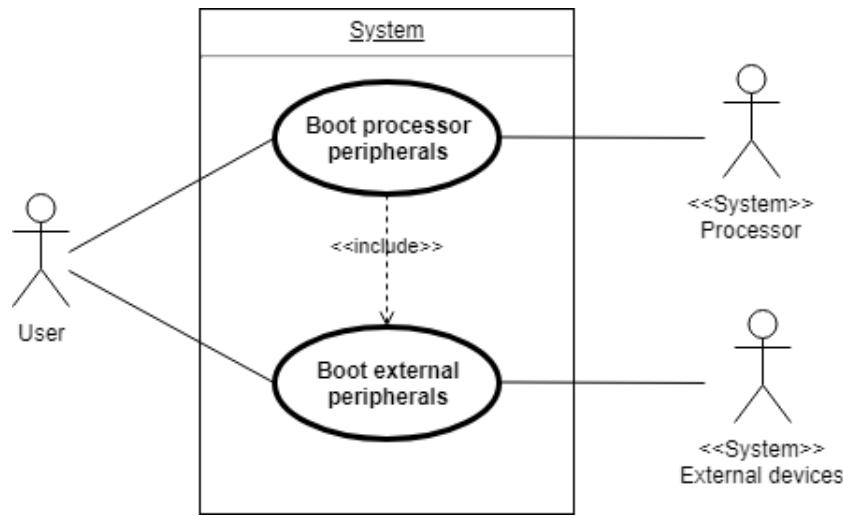
# **Chapter 3**

## **Requirements**

## 3.1 Functional Requirements

This section lists the functional requirements of Ranger Vervet. When appropriate an UML Use Case Diagram is used to graphically represent the requirement.

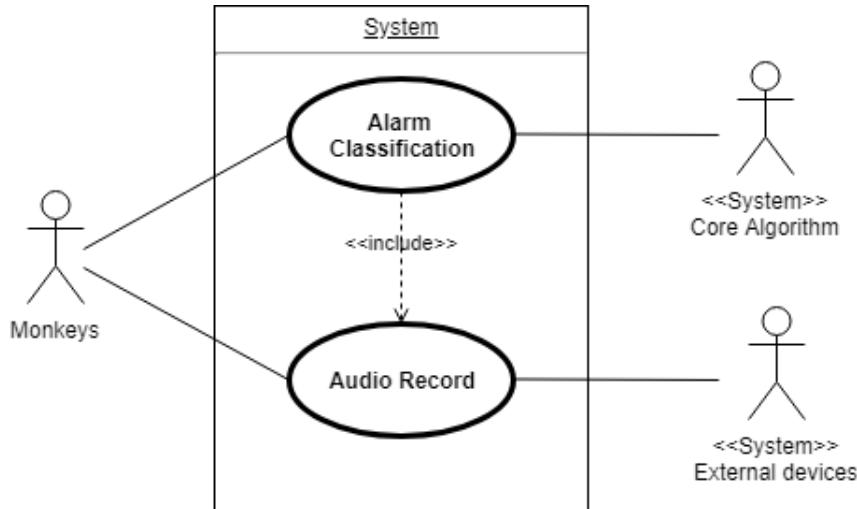
- **Func-Requirement 1:** in booting phase Ranger Vervet has to evaluate the working condition (e.g. component ok or unavailable) of the external devices from which it can obtain any information about their state.



**Figure 3.1:** UML Use Case Diagram for Requirement 1.

- **Func-Requirement 2:** while running, Ranger Vervet has to check every two minutes the working condition (e.g. component ok or unavailable) of the external devices from which it can obtain any information about their state.
- **Func-Requirement 3:** Ranger Vervet has to control its supply level both in booting and running modes, in the latter every two minutes jointly the components check, in order to maximize its availability reducing sudden power issues.
- **Func-Requirement 4:** Ranger Vervet has to be able to put itself in minimum functionality mode when any critical issue arises from its hardware and software components (e.g. broken hardware, unrecoverable software error).
- **Func-Requirement 5:** Ranger Vervet has to process the audio chunks in order to recognize and classify a possible Vervet Monkey alarm call according to the three categories:

- Aerial threat alarm call;
- Land threat alarm call;
- Snake threat alarm call.



**Figure 3.2:** UML Use Case Diagram for Requirement 6.

- **Func-Requirement 6:** Ranger Vervet has to keep on analyzing the recorded audio until an alarm is detected;
- **Func-Requirement 7:** Ranger Vervet has to be able to detect the direction of the incoming alarm call, if detected;
- **Func-Requirement 8:** if an alarm occurs, Ranger Vervet has to properly handle it according to different procedures:
  - Aerial threat alarm call: produce a luminous signal and a high frequency audio signal to scare away the possible aerial predator;
  - Land threat alarm call: produce a luminous signal and a high frequency audio signal to scare away the possible land predator. Take a photo in the direction of the alarm and send it via GPRS to check if there is a human intrusion;
  - Snake threat alarm call: produce a low frequency audio signal to scare away the possible snake.
- **Func-Requirement 9:** after a fixed amount of time dedicated to handle the detected alarm, Ranger Vervet has to return in running mode recording and analyzing the input audio.

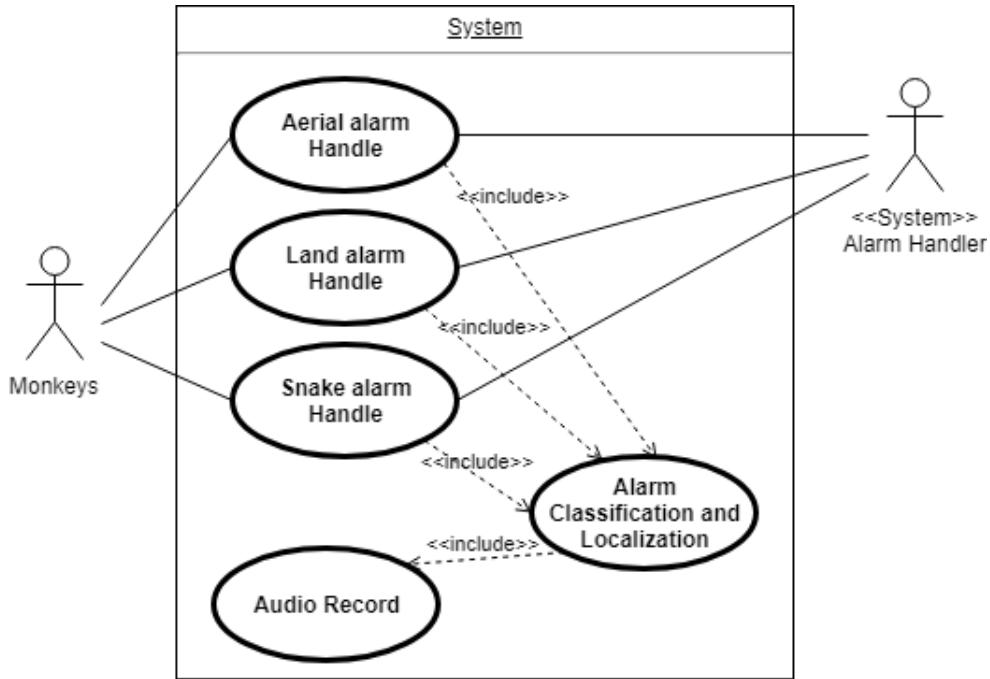


Figure 3.3: UML Use Case Diagram for Requirement 8.

- **Func-Requirement 10:** Ranger Vervet must have a backup alarm notification via SMS in case GPRS is not available.

## 3.2 Non-Functional Requirements

- **NonFunc-Requirement 1:** Ranger Vervet must work only when its hardware components are ok, to maximize availability.
- **NonFunc-Requirement 2:** Ranger Vervet has to record the environmental audio in 0.5sec chunks.
- **NonFunc-Requirement 3:** Ranger Vervet must be able to classify an audio input in an amount of time no longer than 20% of the recorded audio.
- **NonFunc-Requirement 4:** Ranger Vervet must reduce power consumption putting its components in a sleep state when unused.
- **NonFunc-Requirement 5:** Ranger Vervet classification algorithm must classify the input with an error probability lower than 40%.

### 3.3 Structural Requirements

- **Struct-Requirement 1:** Ranger Vervet must have multiple microphones to sense the environmental audio signal.
- **Struct-Requirement 2:** Ranger Vervet must have an audio speaker to produce high frequency and low frequency audio signals to scare predators.
- **Struct-Requirement 2:** Ranger Vervet must have a light signaling device to scare predators.
- **Struct-Requirement 3:** Ranger Vervet must have a camera to take a picture in case of terrestrial alarm to let third parties verify if any poacher or malicious person threatens the animals.
- **Struct-Requirement 4:** Ranger Vervet must have an integrated GSM module to send taken pictures and communicate with third parties.
- **Struct-Requirement 5:** Ranger Vervet must have a stepper motor to steer the camera in the direction of the audio signal when an alarm is detected.

# **Chapter 4**

## **Hardware Specifications**

## 4.1 Introduction

The Ranger Vervet system makes up of numerous hardware elements. Considering as elementary, and so less significant, basic circuit components such as resistors, capacitor, connection wires and so on, the most important are:

- STM32F401RE MCU Board as elaboration centre;
- SIMCOM 800L GSM Module;
- OmniVision OV7670 Camera;
- KY-037 Sound Detector;
- Adafruit NeoPixel FeatherWing LED Matrix;
- Audio Speaker;
- 28BYJ-48 Stepper Motor.

A section is dedicated to each of them, underlining the main features and their role in the overall hardware and software architecture. For more detailed info about these components please refer to the related datasheets in the references section. Figure 4.1 is a high level view of the hardware architecture of the system that stresses the connection between the components and the communication interfaces.

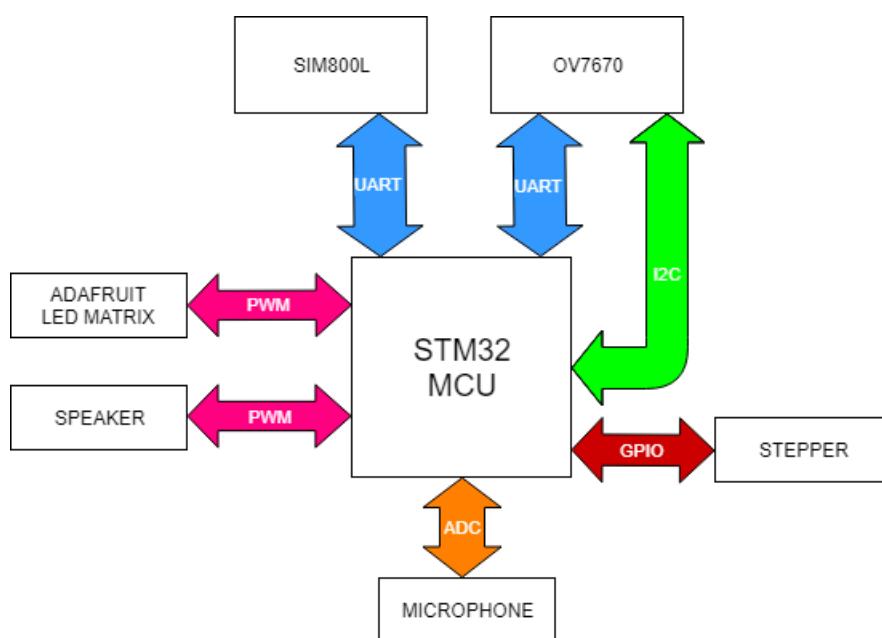
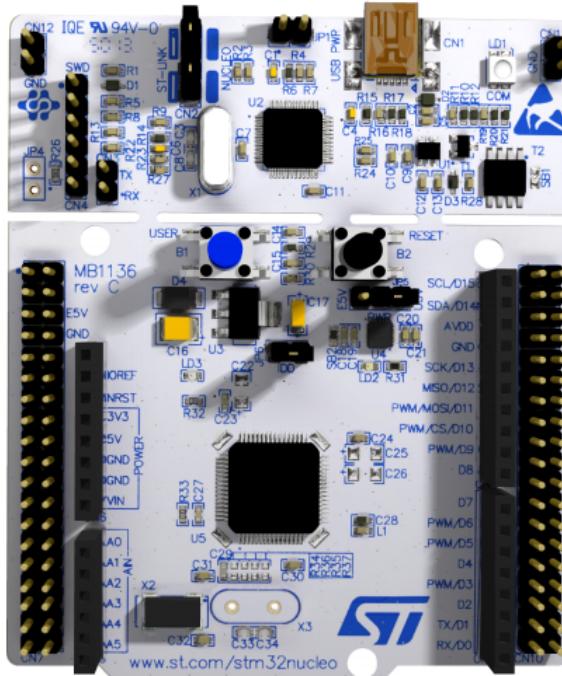


Figure 4.1: Block level hardware architecture.

## 4.2 STM32F401RE Microcontroller Unit



**Figure 4.2:** STMF401RE board.

STMicroelectronics is a huge participant in the MCU market, and offers a vast portfolio of commercial choices to satisfy every need. Ranger Vervet relies on the ARM Cortex-M4-based STM32F4 MCU series that leverages ST's NVM technology and ART Accelerator to reach the industry's highest benchmark scores for Cortex-M-based microcontrollers with up to 225 DMIPS/608 CoreMark executing from Flash memory at up to 180 MHz operating frequency. With dynamic power scaling, the current consumption running from Flash ranges from 89  $\mu$ A/MHz on the STM32F401 up to 260  $\mu$ A/MHz on the STM32F439. The STM32F4 series consists of eight compatible product lines, and Ranger Vervet exploits the STM32F401RE from the STM32F401 line.

The STM32F401RE devices (see figure 4.2) are based on the high-performance ARM Cortex-M4 32-bit RISC core operating at a frequency of up to 84 MHz. Its Cortex-M4 core features a Floating point unit (FPU) single precision which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. The STM32F401xRE incorporate high-speed embedded memories (512 Kbytes of Flash memory, 96 Kbytes of SRAM), and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two

AHB buses and a 32-bit multi-AHB bus matrix. All devices offer one 12-bit ADC, a low-power RTC, six general-purpose 16-bit timers including one PWM timer for motor control, two general-purpose 32-bit timers. They also feature standard and advanced communication interfaces. Following the key features of the board.

- Core: ARM 32-bit Cortex-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator) allowing 0-wait state execution from Flash memory, frequency up to 84 MHz, memory protection unit, 105 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions;
- Memories
  - up to 512 Kbytes of Flash memory
  - up to 96 Kbytes of SRAM
- Clock, reset and supply management
  - 1.7 V to 3.6 V application supply and I/Os
  - 4-to-26 MHz crystal oscillator
  - Internal 16 MHz factory-trimmed RC
  - 32 kHz oscillator for RTC with calibration
  - Internal 32 kHz RC with calibration
- Power consumption
  - Run:  $146 \mu\text{A}/\text{MHz}$  (peripheral off)
  - Stop (Flash in Stop mode, fast wakeup time):  $42 \mu\text{A}$  Typ @  $25^\circ\text{C}$ ;  $65 \mu\text{A}$  max @ $25^\circ\text{C}$
  - Stop (Flash in Deep power down mode, fast wakeup time): down to  $10 \mu\text{A}$  @  $25^\circ\text{C}$ ;  $30 \mu\text{A}$  max @ $25^\circ\text{C}$
  - Standby:  $2.4 \mu\text{A}$  @ $25^\circ\text{C}$  /  $1.7 \text{ V}$  without RTC;  $12 \mu\text{A}$  @ $85^\circ\text{C}$  @ $1.7 \text{ V}$
  - VBATsupply for RTC:  $1 \mu\text{A}$  @ $25^\circ\text{C}$
- 1x12-bit, 2.4 MSPS A/D converter: up to 16 channels
- General-purpose DMA: 16-stream DMA controllers with FIFOs and burst support

- Up to 11 timers: up to six 16-bit, two 32-bit timers up to 84 MHz, each with up to four IC/OC/PWM or pulse counter and quadrature (incremental) encoder input, two watchdog timers (independent and window) and a SysTick timer
- Debug mode
- Up to 81 I/O ports with interrupt capability
  - Up to 78 fast I/Os up to 42 MHz
  - All I/O ports are 5 V-tolerant
- Up to 12 communication interfaces
  - Up to 3 x I2C interfaces (SMBus/PMBus)
  - Up to 3 USARTs (2 x 10.5 Mbit/s, 1 x 5.25 Mbit/s), ISO 7816 interface, LIN, IrDA, modem control)
  - Up to 4 SPIs (up to 42Mbit/s at fCPU= 84 MHz), SPI2 and SPI3 with muxed full-duplex I2S to achieve audio class accuracy via internal audio PLL or external clock
  - SDIO interface
  - Advanced connectivity: USB 2.0 full-speed device/host/OTG controller with on-chip PHY
- CRC calculation unit
- 96-bit unique ID
- RTC: subsecond accuracy, hardware calendar
- All packages are ECOPACK2

For precise and complete information about the board, please refer to the official documents [3], [4], [5].

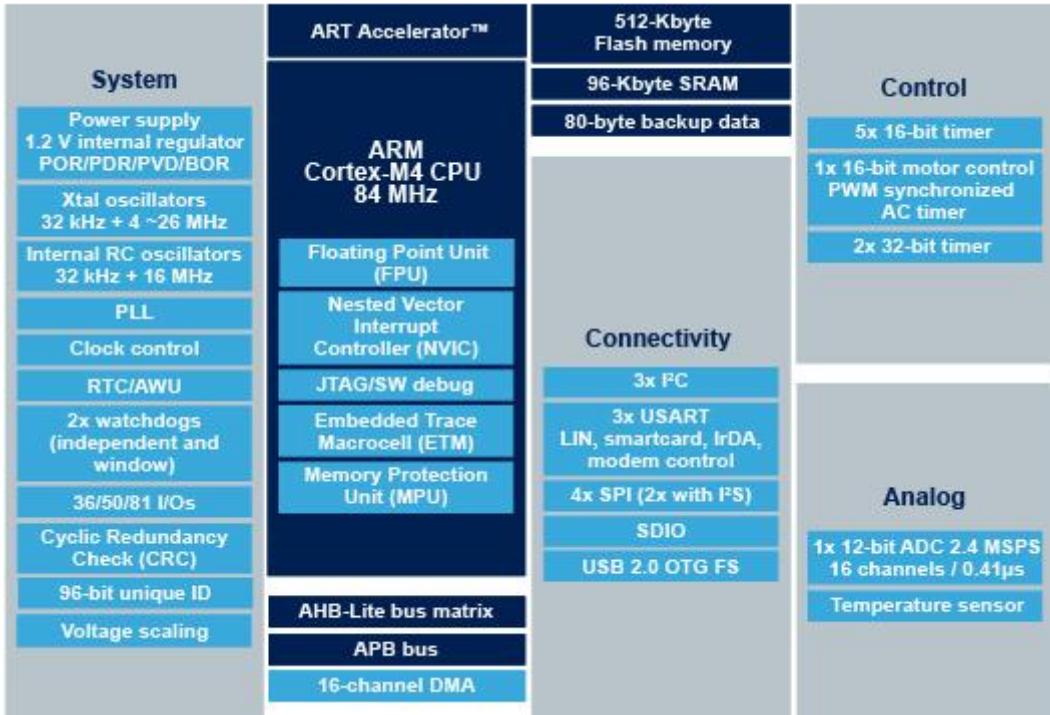


Figure 4.3: STMF401RE circuit diagram.

### 4.3 SIMCOM 800L GSM Module

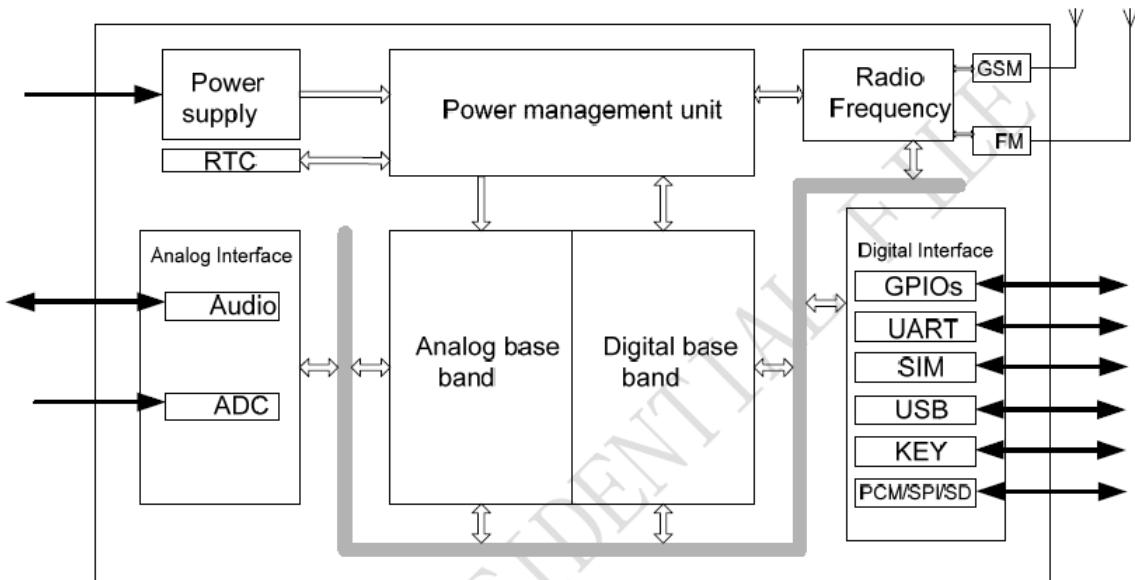


Figure 4.4: SIMCOM 800L GSM Module.

SIMCOM 800L is a complete GSM development board that allows users to take advantage of all the services provided by the GPRS/EGPRS network infrastructure, calls, SMS, Internet access, with a standardized interface based on the AT protocol.

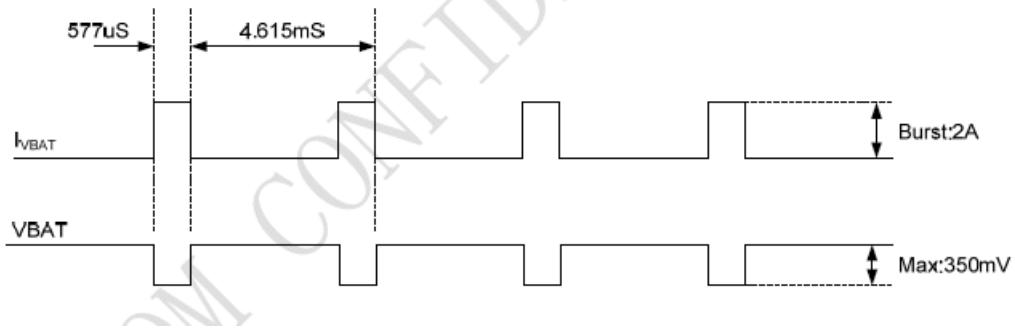
SIMCOM 800L is a quad-band GSM/GPRS module that supports the four frequencies bands GSM850MHz, EGSM900MHz, DCS1800MHz and PCS1900MHz, so to be worldwide usable. Supports multi-slotted GPRS with four possible coding schemes, CS-1 CS-2 CS-3 and CS-4. By varying them the data transfer rate scales from 9.05kbps to 85.6kbps. Supports different codecs and echo and noise cancellation to improve the calling experience. The module pinout is presented in table 4.1.

Type	Verse	Name	Description
Analog	IN	NET	Pin reserved to solder the helical antenna of the module
Supply	-	VCC	Supply voltage between 3.4V to 4.4V
Digital	IN	RST	Hard Reset signal
Digital	IN	RXD	Serial interface receiver
Digital	OUT	TXD	Serial interface transmitter
Supply	-	GND	Ground reference
Digital	OUT	RING	Ring indicator for calls ad SMS
Digital	IN	DTR	Sleep mode activation/deactivation
Digital	IN	MIC+	Differential microphone input
Digital	IN	MIC-	Differential microphone input
Digital	OUT	SPK+	Differential speaker interface
Digital	OUT	SPK-	Differential speaker interface

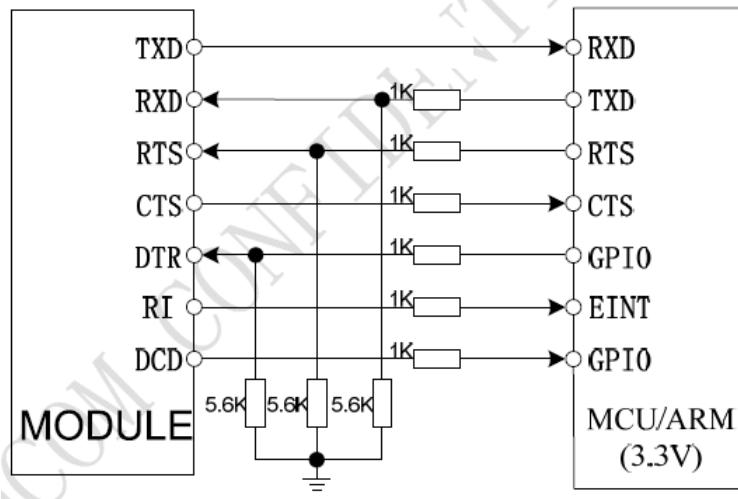
**Table 4.1:** SIMCOM 800L GSM Module pinout.**Figure 4.5:** SIMCOM800L GSM Module functional diagram.

Power supply can be a very critical issue when using this device, because it requires a very stable voltage source (recommended 4V) able to tolerate the current requirements of the module. Indeed during data transmission, or during network connection, SIMCOM 800L can drain high currents up to 2A that cause a significant voltage drop, see figure 4.6. If the power source cannot stay up to 3.4V during these short time intervals, the module may abruptly shut down and restart. Using the AT protocol is also possible to set the GSM board in sleep mode or in power saving mode to reduce resources consumption.

The serial communication with SIMCOM 800L is fundamental in order to use it as modem



**Figure 4.6:** SIMCOM 800L power supply diagram. VBAT is the GSM chip pin used for the power supply.



**Figure 4.7:** SIMCOM800L GSM Module serial connection. This scheme refers to the connection with a DTE using 3.3V input/output.

and control it via AT commands, and is based on UART, with the module acting as DCE (Data Communication Equipment) and the MCU as DTE (Data Terminal Equipment). Also in this case the module is very sensitive to voltage levels, as indicated in table 4.2, and the producer suggests the correct circuit connections, figure 4.7 and [6]. A very interesting feature of the serial interface is the autobauding functionality that supports different baud rates up to 57600bps.

Name	Min	Max
V0H	2.7V	2.8V
V0L	0V	0.1V
V1H	2.5V	2.8V
V1L	0V	0.3V

**Table 4.2:** SIMCOM 800L GSM serial interface voltage ratings.

SIMCOM800L accepts SIM cards that comply with GSM1 specification and GSM2+ specification, and many of the modern SIM cards are retro-compatible with these standards. It

comes out with an integrated SIM card holder for microSIM. Many other connection interfaces are available, for keypads, LCD monitors, microphones, speakers, and all that is necessary to essentially build a 2G phone around this module, for more interesting features refer again to [6].

### 4.3.1 SIMCOM800L GSM Module Hardware - Software Interface

The module usage is quite straightforward because it is essentially an exchange of AT messages between DCE and DTE via UART, that allow clients to perform both "hardware" and "service" requests, such as turning on and off the module, make a call, establish a TCP/IP connection via GPRS and more. For a complete view on the AT commands supported by SIM800L refer to [7] and [8]. After the initialization and connection to GPRS, Ranger Vervet puts the module in sleep mode and awakes it when a terrestrial alarm is detected. At this point the captured image is sent to a third party web server, as body of a HTTP POST request relying on the GPRS network infrastructure. When any Internet related error occurs, a warning message about the alarm is sent via SMS, as alarm notification backup procedure.

## 4.4 OmniVision OV7670 Camera Module

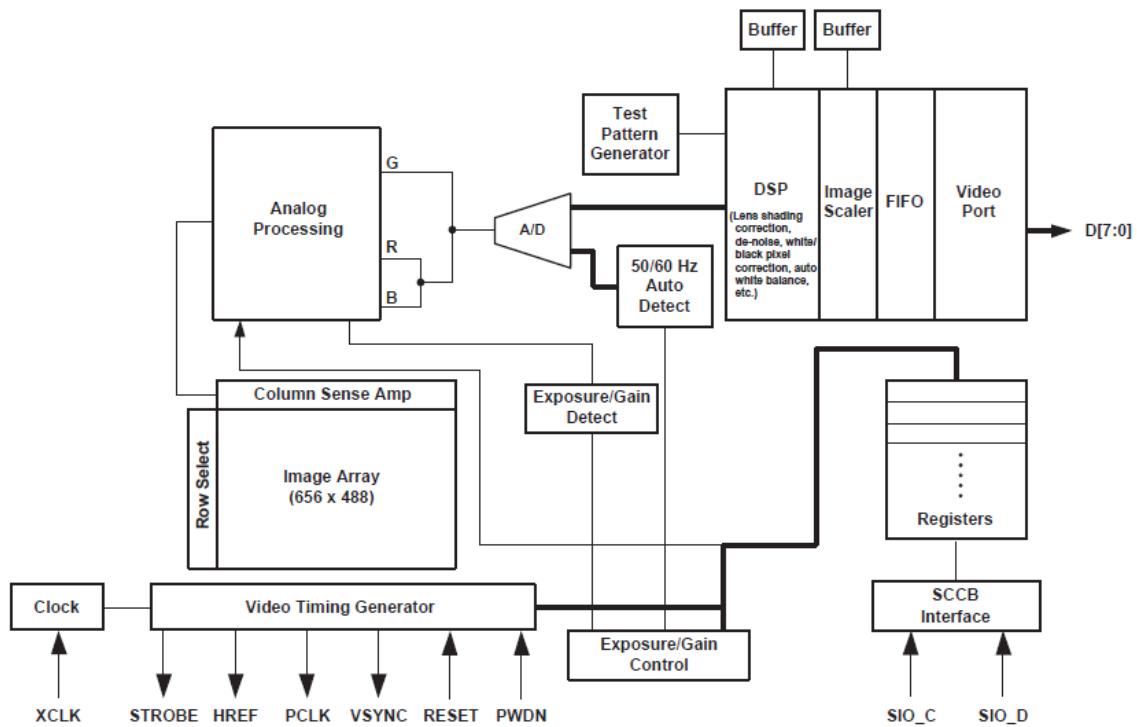


**Figure 4.8:** OmniVision OV7670 Camera Module.

OV7670 is a low voltage CMOS image sensor that acts as VGA camera capable of reaching a 30fps image transfer rate in VGA resolution. This module supports a large amount of image preprocessing functionalities to improve the image quality, many of them that can be set in automatic mode. Some of the most valuable features follow:

- support for VGA and CIF resolutions, and lower version (eg QVGA, QCIF, QQVGA ecc.) in RGB, YUV and YCrBr image formats;
- automatic image control functions, Automatic Exposure Control (AEC), Automatic Gain Control (AGC), Automatic White Balance (AWB), Automatic Band Filter (ABF), Automatic Black-Level Calibration (ABLC);
- image quality control including color saturation, hue, gamma, sharpness;
- support for flash mode;
- support for image scaling;
- saturation level auto adjustment;
- auto edge enhancement;
- auto de-noising.

To offer all this, the OV7670 module has a quite complex architecture, shown in figure 4.9, that is next briefly explained For complete information refer to [9], [10], [11]. The main com-



**Figure 4.9:** OmniVision OV7670 Camera Module hardware architecture.

ponent is the 656x488 pixels image sensor array, of which 640x480 are the active pixels. The

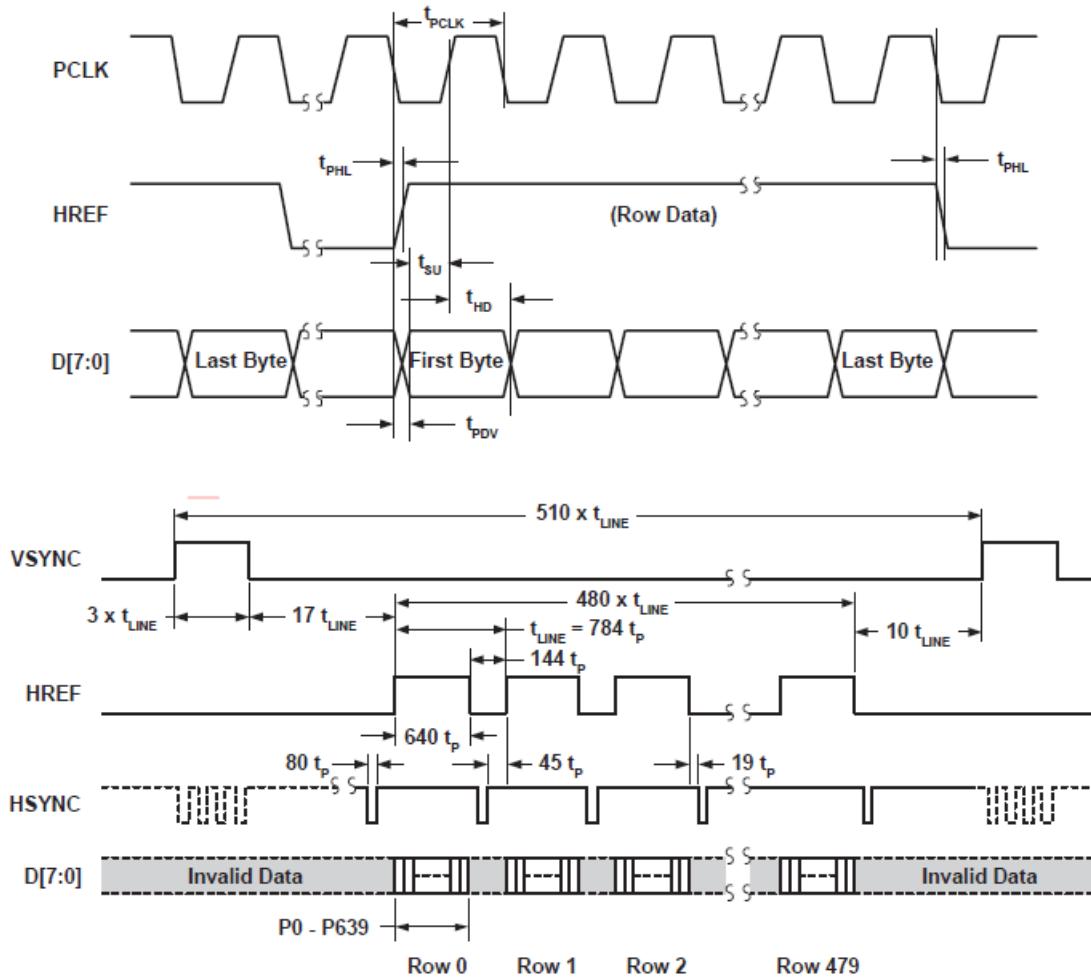
analog signal processor, A/D converters and the DSP processor are responsible of elaborating the image to apply all the image processing features aforementioned. A functional block is dedicated to the scaling of the image in order to change its resolution, also out of the standard supported resolutions. The camera behavior is controlled by setting its 201 internal registers, which can be accessed via the Serial Camera Control Bus (SCCB) interface [12], a specific protocol defined by the OmniVision vendor, that is however compatible with I2C.

The camera is fully based on timing signals because it has no memory elements. It has to be fed with an external clock between 10 and 24 MHz in order to be working. This signal can be pre-scaled or increased by PLL according to the registers setting to satisfy the user timing requirements.

The output signals produced by the camera, necessary to read frames, are synchronized on the basis of the input clock. In particular this module provides two fundamental synchronization signals for vertical and horizontal image reference, and a pixel clock output signal to capture at the right time each pixel. For the absence of memory elements the captured image is sent as a bit stream pixel by pixel throughout 8 output pins. The timing diagrams in 4.10 show these concepts and table 4.3 indicates the complete 18 pins pinout.

Type	Verse	Name	Description
Supply	-	VCC Address	Supply voltage with maximum rating of 3.5Volts
Supply	-	GND	Ground reference
Digital	INOUT	SIOD	SCCB interface data line
Digital	IN	SIOC	SCCB interface clock line
Digital	IN	XCLK	Clock source, 10MHz to 24MHz
Digital	OUT	VSYNC	Vertical synchronization
Digital	OUT	HREF	Horizontal reference
Digital	OUT	PCLK	Pixel clock reference
Digital	IN	RST	Reset camera registers
Digital	IN	PWDN	Set camera in stand-by mode
Digital	IN	D0	Pixel byte bit 0
Digital	IN	D1	Pixel byte bit 1
Digital	IN	D2	Pixel byte bit 2
Digital	IN	D3	Pixel byte bit 3
Digital	IN	D4	Pixel byte bit 4
Digital	IN	D5	Pixel byte bit 5
Digital	IN	D6	Pixel byte bit 6
Digital	IN	D7	Pixel byte bit 7

**Table 4.3:** OV7670 Camera Module pinout.



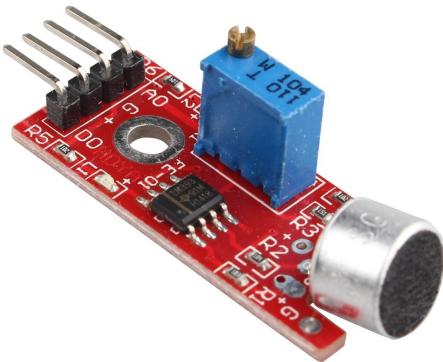
**Figure 4.10:** OV7670 camera module timing diagram. In the first diagram it is shown that each byte has to be read on the rising edge of the PCLK signal, and only when HREF is set. Depending on the color format a variable number of consecutive bytes constitutes a single pixel. In the second diagram, supposing a VGA resolution, it is shown how when VSYNC completes a pulse, HREF has 480 pulses, one per each image row. During each of these pulses 640 pixels of the row have to be read.

#### 4.4.1 OV7670 Camera Module Hardware-Software interface

Ranger Vervet sets its OV7670 in RGB565 color mode, so each pixel is 16bits deep. The resolution format is QQVGA, so a complete frame is composed by 160x120 pixels, for a total of 38400 bytes. The timing protocol to read the frame is bit-banged, the MCU reads the synchronization signals and at the right time captures the byte representing half a pixel. PCLK evolution is, using camera settings, limited to the time interval in which also HREF is high. This means, please observe figure 4.10, that the horizontal reference signal is of no use and Ranger Vervet reads only VSYNC to synchronize itself on the line of the frame and PCLK to retrieve the image data. Ranger Vervet keeps OV7670 in sleep mode until a terrestrial alarm

is detected, only in this case the camera is awaken and its output read to capture a picture.

## 4.5 KY-037 Microphone Sensor Module



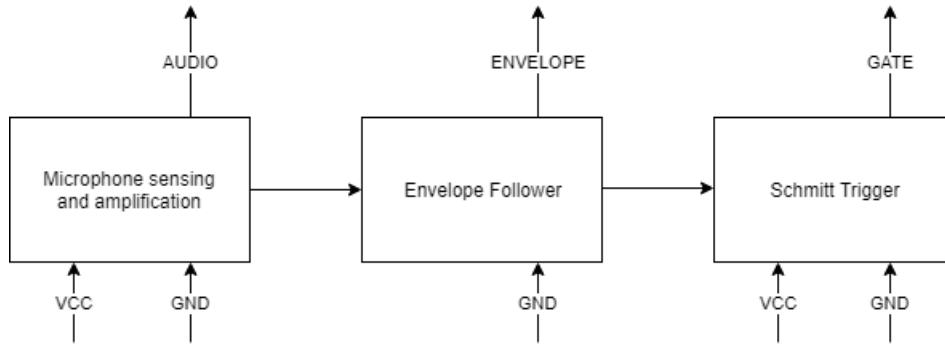
**Figure 4.11:** KY-037 Microphone Sensor Module

KY-037 is a sound detector device, a simple IC board with a microphone and some auxiliary circuitry. It has the pinout reported in table 4.4.

Type	Verse	Name	Description
Supply	-	VDD	Power supply voltage between 3.5 and 5.5 Volts
Supply	-	GND	Ground reference
Analog	OUT	AUDIO	Audio output of the sound detector
Analog	OUT	ENVELOPE	Envelope output tracing the amplitude of the sound
Digital	OUT	GATE	Indicates if a sound has been detected or not

**Table 4.4:** KY-037 sound detector pinout.

Without going into its exact electrical schematic, which can be found in [13], its functionalities can be sum up, from a high level perspective, in the next block diagram scheme of figure 4.12. The sensitivity of the device can be regulated by changing the resistance on the terminals of the amplifier connected to the physical microphone, turning a screw mounted on the board. It is also good to remember that this sensor is intrinsically analog, so it suffers from mechanical noise and electrical noise, so its measurements may be inaccurate in some working conditions.



**Figure 4.12:** KY-037 sound detector functional block diagram. The first block contains the physical microphone, and after some amplification of the small voltage that produces, drives the AUDIO output. The second element is a rectifier circuit that detects the input signal peaks to generate the ENVELOPE output. In the end, the last block is a Schmitt Trigger thresholding that toggles the GATE output.

#### 4.5.1 KY-037 Hardware-Software interface

Ranger Vervet uses three KY-037 in order to sense the environmental audio. Their analog output AUDIO is sampled at a frequency of 10kHz by the MCU embedded ADC using three channels. Then a quantized value is obtained with the maximum ADC allowed resolution of 12bits. ADC conversion procedure is triggered by a timer and the three converted values are transferred in a memory buffer via DMA. Then a simple algorithm, see 5.3.2, decides which of the samples has to be saved in the larger buffer (5000 elements, given the sampling frequency of the ADC) where the audio recording of about 0.5sec is, conversion by conversion, built up. This procedure takes place in the ISR for ADC sampling completion.

### 4.6 Adafruit NeoPixel FeatherWing LED Matrix

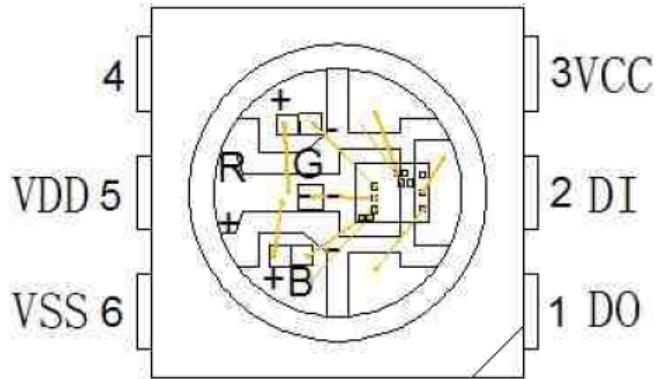


**Figure 4.13:** NeoPixel ws2812 pin diagram.

Adafruit NeoPixel FeatherWing is a 4x8 RGB LED Matrix suitable for any kind of applica-

tion based on light signaling, thanks to a particular hardware structure. This component is the union of 32 single Adafruit NeoPixel LEDs, also known as ws2812. Controlling the LED matrix simply means to control a group of ws2812.

Each LED has 6 pins, one of which unused, indicated in table 4.5, and offers the possibility to be connected in a daisy chain, where the output of a led becomes the input of the successive. In this manner a single control signal can program a theoretically infinite series of ws2812.

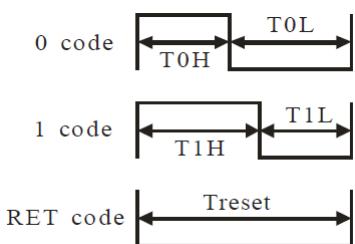


**Figure 4.14:** NeoPixel ws2812 pin diagram.

Type	Verse	Name	Description
Supply	-	VCC	Power supply control circuit
Supply	-	VDD	Power supply LED
Supply	-	VSS	Ground reference
Digital	IN	DI	Control data signal input
Digital	OUT	DO	Control data signal output

**Table 4.5:** NeoPixel ws2812 pinout.

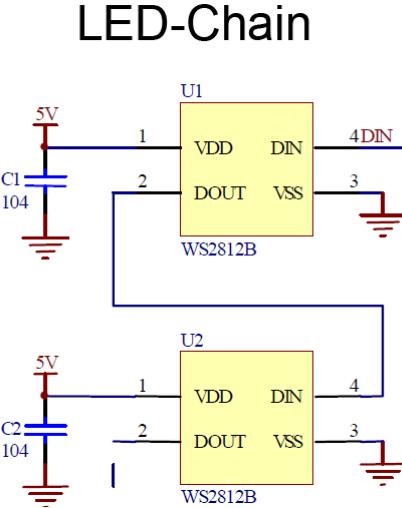
A ws2812 can be programmed by sending a signal in which the desired RGB color is coded in 3 bytes, setting each bit high or low value according to the timing protocol provided by the datasheet in [14], and for commodity shown in figure 4.15.



Signal	Meaning	Transfer Time
T0H	0 code high voltage	$0.35\mu s$
T0L	0 code low voltage	$0.7\mu s$
T1H	1 code high voltage	$0.8\mu s$
T1L	1 code low voltage	$0.6\mu s$
RST	reset low voltage	above $50\mu s$

**Figure 4.15:** ws2812 commands timing diagram.

**Table 4.6:** NeoPixel ws2812 timing specifications.



**Figure 4.16:** ws2812 daisy chaining.

Using the daisy chain scheme in figure 4.16 is possible to set the color of the Adafruit NeoPixel FeatherWing LEDs, just supplying the module with the right voltage level and commanding it with a signal at the DIN pin that codes the RGB color of each LED [15]. Each ws2812 buffers its 24 bits and sends the remaining to the following LED at its DO pin after internal signal reshaping to reduce distortion. To improve the stability of the coloration some delays could be useful in order to compensate the circuitry delay due to the signal transmission and manipulation over the LED chain.

#### 4.6.1 AdaFruit NeoPixel FeatherWing Hardware-Software interface

The timing protocol of this device is simulated using bit banging, in particular sending via PWM a square wave encoding the information about each led coloration. A simple array contains two possible values that encode high and low values of the ws2812 timing diagram, as in figure 4.15, for each bit of the 24bits string representing the required color. This is repeated for each LED in the matrix, with the addition of one fictitious LED to deal with some delay issues. The result is a binary message that encodes the entire matrix coloration, which is send to the external device by PWM generation in DMA mode.

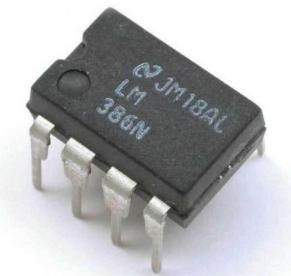
This LED matrix is light up when the system detects an alarm for aerial threat or land threat, in such a way to produce a repetitive white flash to scare away the attacker. This task is accomplished generating an interrupt with a timer and turning alternatively on and off the device during the corresponding ISR.

## 4.7 Audio Speaker



**Figure 4.17:** Ranger Vervet audio speaker

Ranger Vervet is equipped with a  $4\Omega$  3W audio speaker, supplied throughout an audio amplifier to compensate the low power output drive of the MCU. The aforementioned amplifier is a common LM386, a low voltage audio amplifier, discussed in [16].

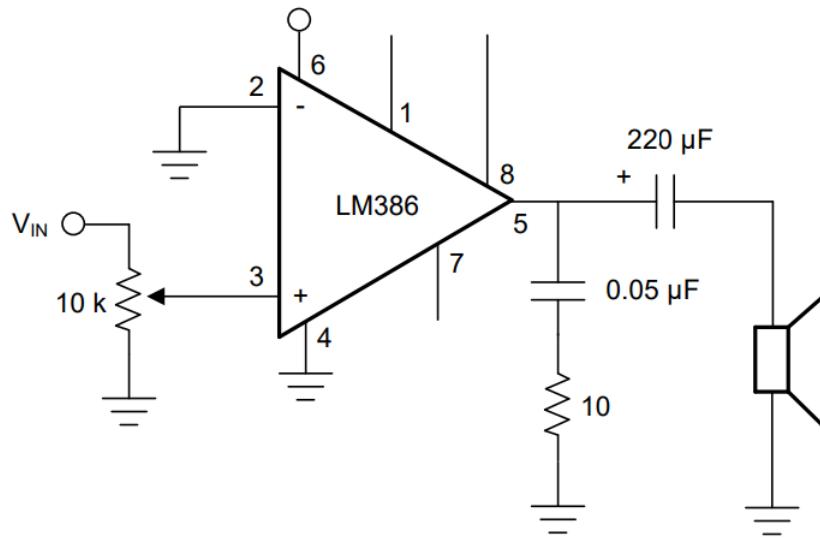


**Figure 4.18:** LM386 audio amplifier.

As suggested by the previously cited document, Ranger Vervet arranges these two components according the following circuit, to obtain a gain of about 20 for the input voltage.

### 4.7.1 Audio Speaker Hardware-Software interface

The audio speaker is commanded with a PWM square wave, connected to pin 3 in figure 4.19. Wave frequency is selected in such a way to produce a low frequency sound, at 200Hz, and a high frequency sound, at 20kHz. Duty cycle is fixed, to half of the wave period. This audio signal is enabled by Ranger Vervet only when an alarm is detected.



**Figure 4.19:** Audio Speaker application circuit. Input number 3 is connected to the input voltage signal, pin number 6 to a 5V voltage supply.

## 4.8 28BYJ-48 Stepper Motor



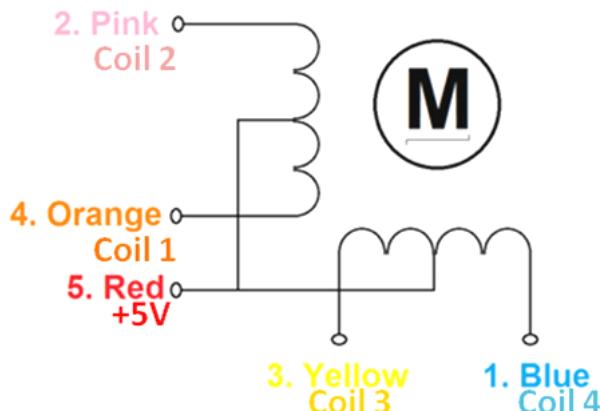
**Figure 4.20:** 28BYJ-48 Stepper Motor.

28BYJ-48 is a brushless motor commonly used for small projects that don't need much speed or torque. It is a unipolar motor that comes out with onboard gear reduction of 1/64, and some of its main features are listed below, and in details in [17]:

- 5V input voltage;

- 4 phases;
- 50Ohm inner resistance;
- 34mN/m torque;
- Stride angle  $5.625^\circ/64$ .

It can be used both in half mode and single mode, with an elementary pulse rotation of respectively  $0.0879^\circ$  and  $0.176^\circ$ . With simple maths it can be shown that a full rotation of  $360^\circ$  can be obtained with 2048 (half mode) or 4096 (single mode) steps. Single mode rotation is preferable to generate more torque because at each step two coils are electrified. The pinout, in table 4.7 is quite simple, and the switching sequence for single mode driving is shown in table 4.8.



**Figure 4.21:** 28BYJ-48 pinout representation using internal motor coils.

Type	Verse	Name	Description
Supply	-	VDD	Power supply control circuit
Digital	IN	COIL1	Coil input voltage
Digital	IN	COIL2	Coil input voltage
Digital	IN	COIL3	Coil input voltage
Digital	IN	COIL4	Coil input voltage

**Table 4.7:** 28BYJ-48 Stepper Motor pinout.

Step	COIL1	COIL2	COIL3	COIL4
1	High	Low	Low	High
2	High	High	Low	Low
3	Low	High	High	Low
4	Low	Low	High	High

**Table 4.8:** 28BYJ-48 Stepper Motor switching sequence.

Because of the too large current amounts involved during the motor activity, 28BYJ-48 is usually connected to a driver board, such as ULN2003. It is a high voltage, high current Darlington array with emitters able to produce 500mA [18]. It also comes out of the box with suppression diodes for managing inductive loads, such as 28BYJ-48. This module has the following pinout.



**Figure 4.22:** ULN2003 stepper motor driver.

Type	Verse	Name	Description
Supply	-	VCC	Power supply control circuit
Supply	-	GND	Ground reference voltage
Digital	IN	IN1	First input voltage
Digital	IN	IN2	Second input voltage
Digital	IN	IN3	Third input voltage
Digital	IN	IN4	Fourth input voltage

**Table 4.9:** ULN2003 motor driver pinout.

#### 4.8.1 28BYJ-48 Hardware-Software interface

In Ranger Vervet System, 28BYJ-48 positions the camera in the direction of the sound extract by the sound localization algorithm in section 5.3.1. This procedure is triggered only when a land threat alarm is detected, otherwise the motor is fixed in steady position. At the end of the alarm handling routine the motor brings back the camera in steady position. Being the stepper motor not equipped with any encoder, there is no way to internally calculate the current position. This drawback comes out during the boot procedure of the system, where in the hardware initialization phase the motor has to position the camera in steady position given the current, unknown, angle of the motor. To solve the issue, in the initialization phase the motor turns until it obscures a photoresistor triggering an interrupt that indicates the line checking and allowing it to turn back in the steady position.

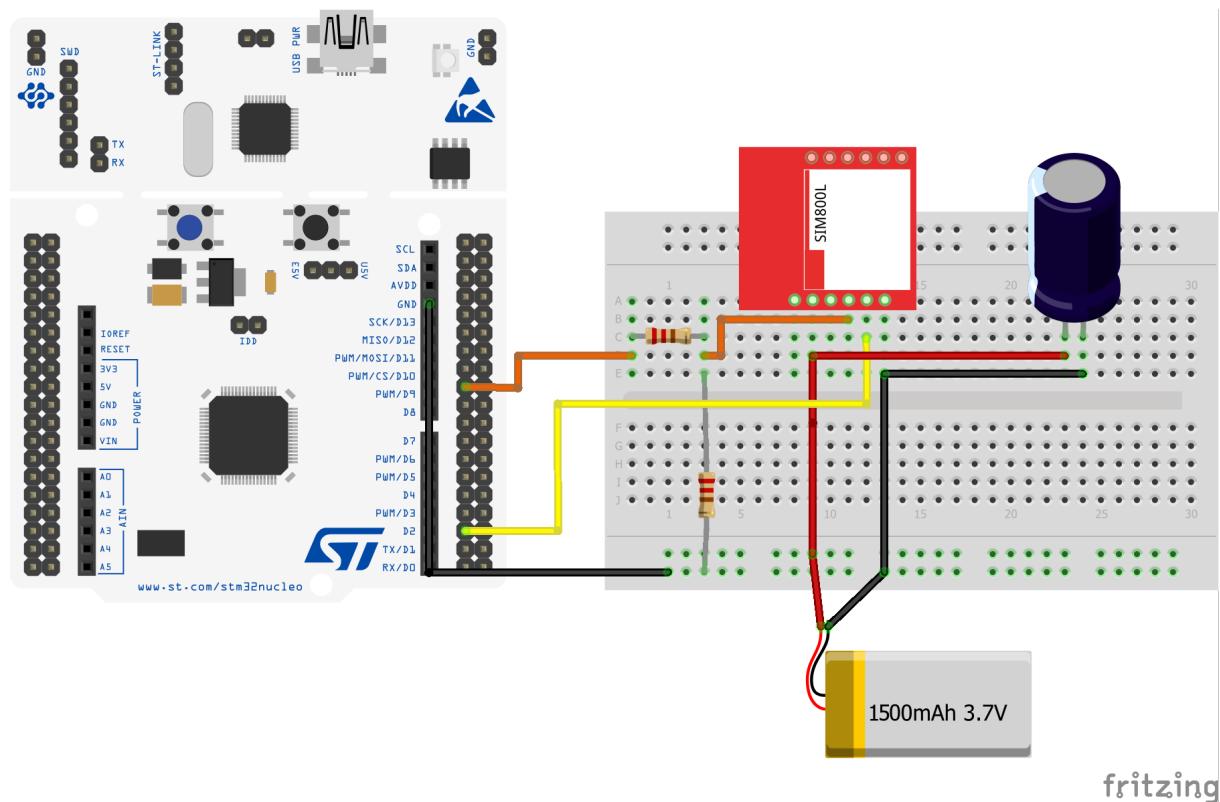
As previously explained, the stepper motor can be controlled just toggling the input pins

according to a fixed sequence, and the photoresistor for the repositioning requires some ADC conversion of the light value read.

## 4.9 Electrical Connections

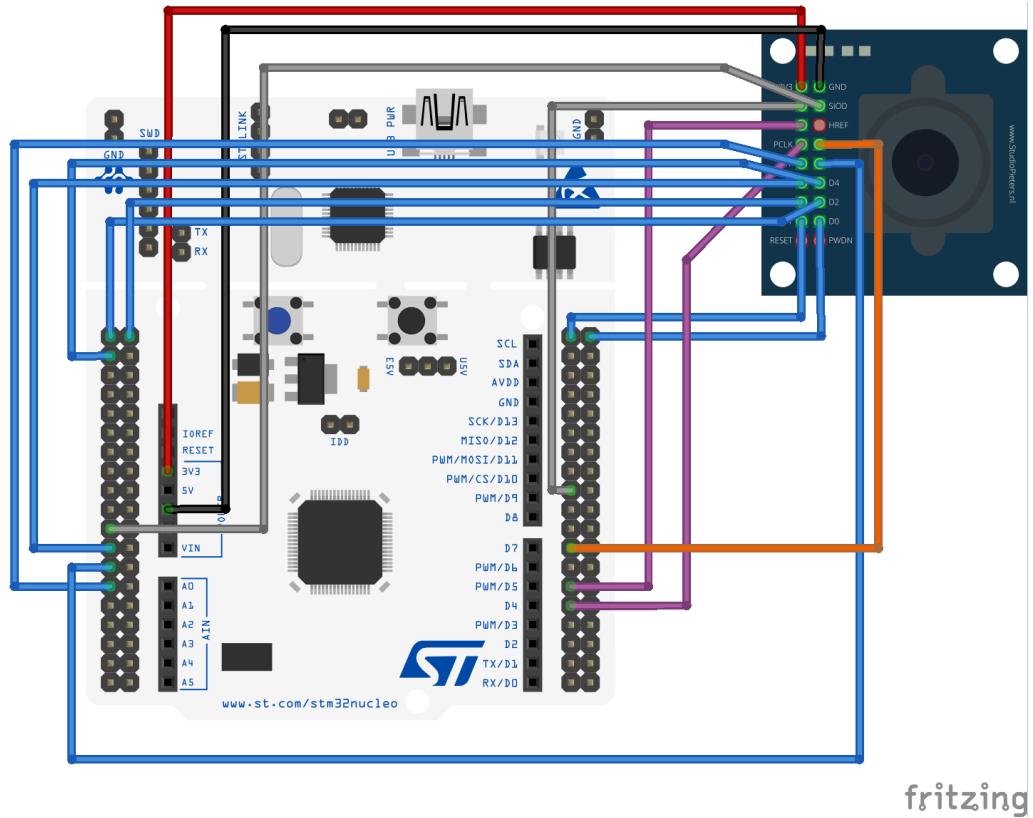
This section provides several schematics to show the interconnection of the various components. For any of the previous device a separate schematic illustrating the connection with the MCU is provided. Please refer to figure 4.29 for the electric schematic of the whole Ranger Vervet System.

### 4.9.1 SIMCOM800L - STM32F401RE connection



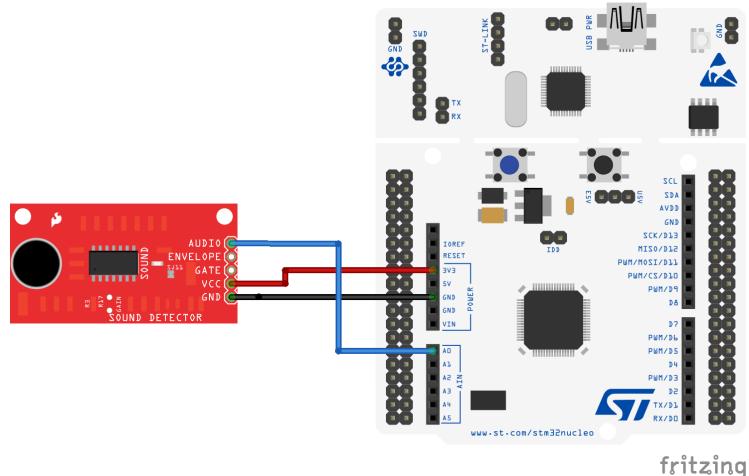
**Figure 4.23:** SIMCOM800L - STM32F401RE connection. GSM board is supplied with a 3.7V LiPo battery at 1500mAh, it has common ground with MCU to grant the correct serial communication. STM32F401RE UART transmitter is located in PB6, and then level shifted with a voltage divider to meet SIMCOM800L specifications. A 1000 $\mu$ F electrolytic capacitor is in parallel with GSM board supply pins to provide additional current during transmission bursts in case the battery cannot provide enough energy.

### 4.9.2 OV7670 - STM32F401RE connection



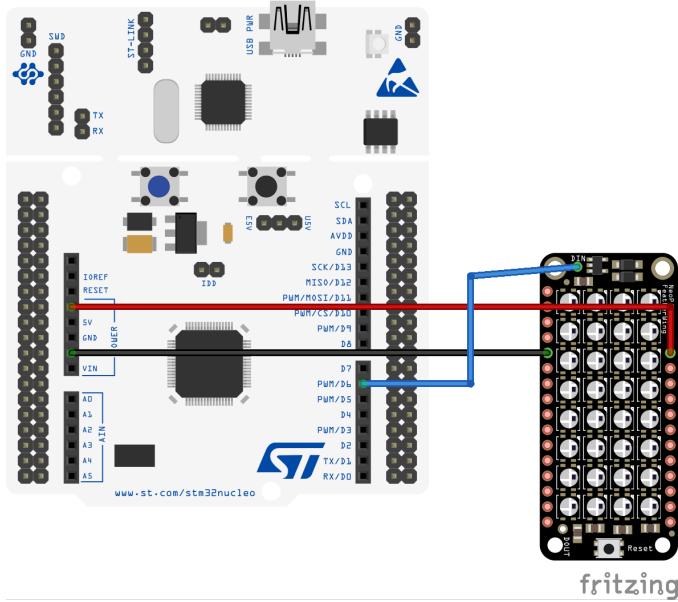
**Figure 4.24:** OV7670 - STM32F401RE connection. Blue wires connect pins D0 to D7 transmitting the image bytes to PC8-PC15 of the board. PA8, in orange, is connected to OV7670 XCLK, purple wires connect VSYNC to PB4 and PCLK to PB5 for data synchronization. Grey wires are reserved to I<sub>2</sub>C SCL and SDA. The camera module is supplied with 3.3V.

### 4.9.3 KY037 - STM32F401RE connection



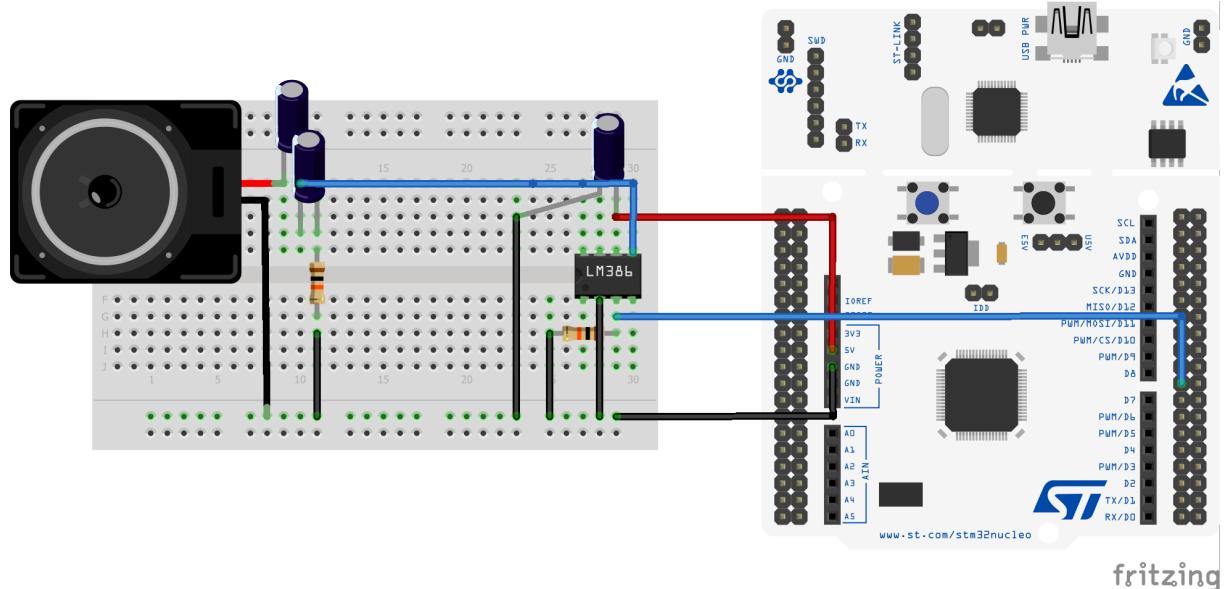
**Figure 4.25:** KY037 - STM32F401RE connection. KY037 microphones need only supply and ground, and then a wire to an analog input of the MCU, as PA0 in figure to capture the AUDIO output.

#### 4.9.4 Adafruit NeoPixel FeatherWing - STM32F401RE connection



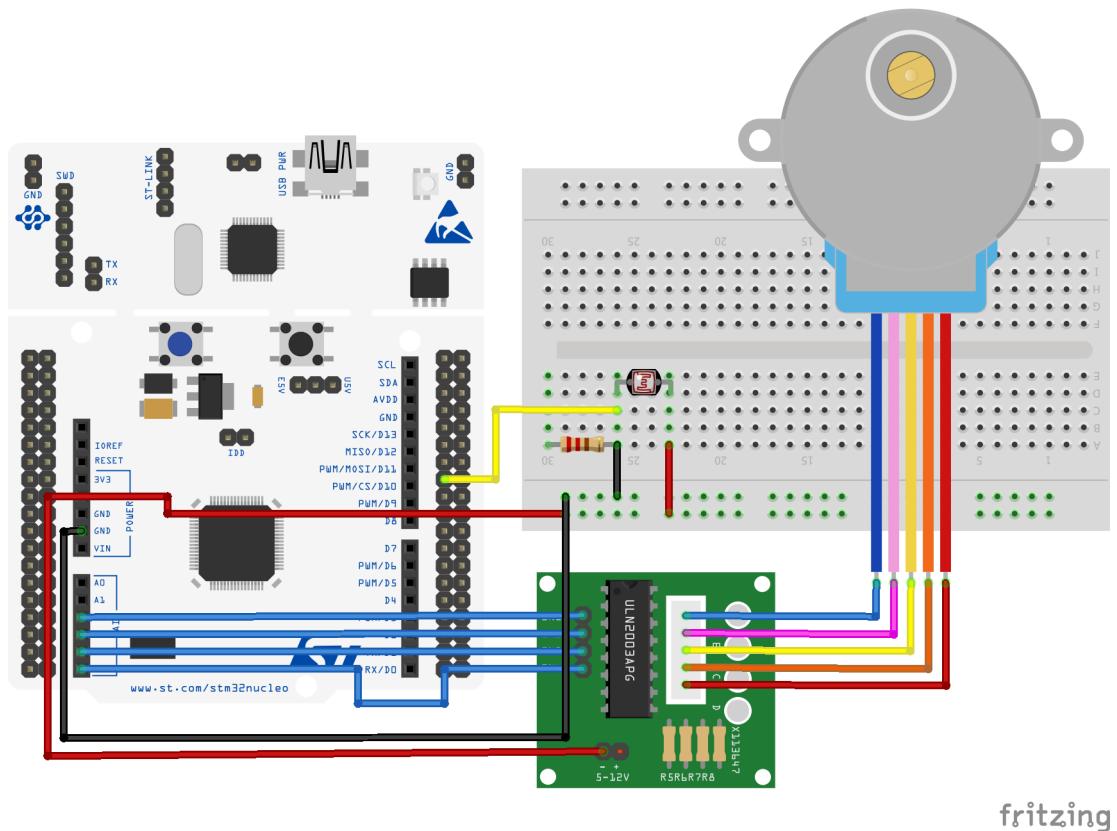
**Figure 4.26:** Adafruit NeoPixel FeatherWing - STM32F401RE connection. Out of the supply pins, at 3.3V, NeoPixel FeatherWing DIN is connected to STM PB10 which is the output of TIM2 CHANNEL3 for PWM generation (blue wire).

#### 4.9.5 Audio Speaker - STM32F401RE connection



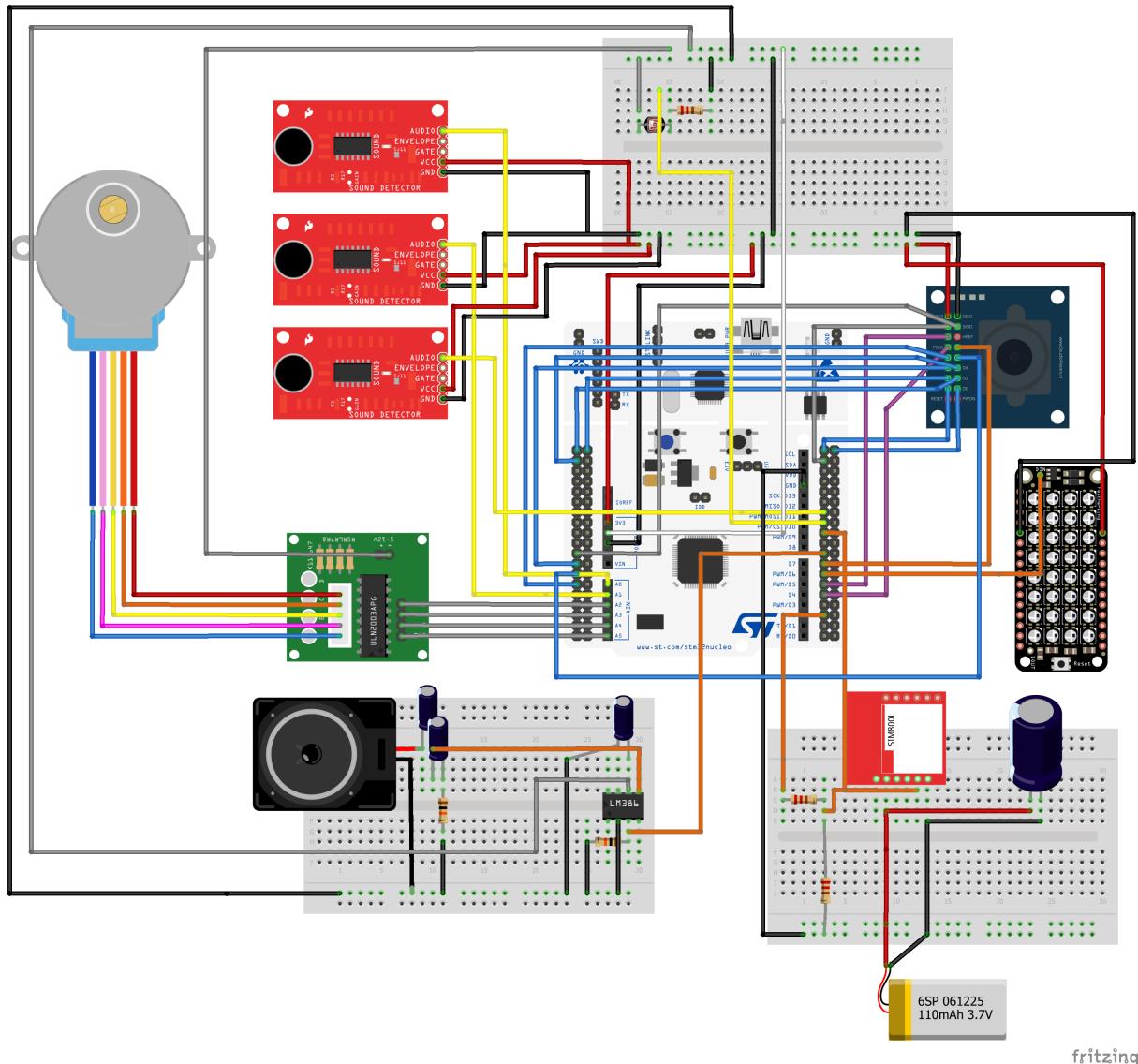
**Figure 4.27:** Audio Speaker - STM32F401RE connection. This circuit implements the one shown in section 4.7, supplying LM386 with PWM modulation incoming from PA9, which is connected to TIM1 CHANNEL2 of STM board. An additional  $10\mu F$  decoupling capacitor is in parallel with the voltage supply to reduce noise.

#### 4.9.6 28BYJ-48 - STM32F401RE connection



**Figure 4.28:** 28BYJ-48 - STM32F401RE connection. Connecting the MCU to the stepper motor is quite straightforward, paying attention to the correct order of the coils pins passing through ULN2003 driver. MCU pins selected for this task are PA4, PB0, PC0, PC1. Power supply comes from 5V MCU output. The schematic also contains the photoresistor, connected to ADC pin PA3, used to emulate a limit switch not available in 28BYJ-48.

#### 4.9.7 Ranger Vervet System wiring diagram



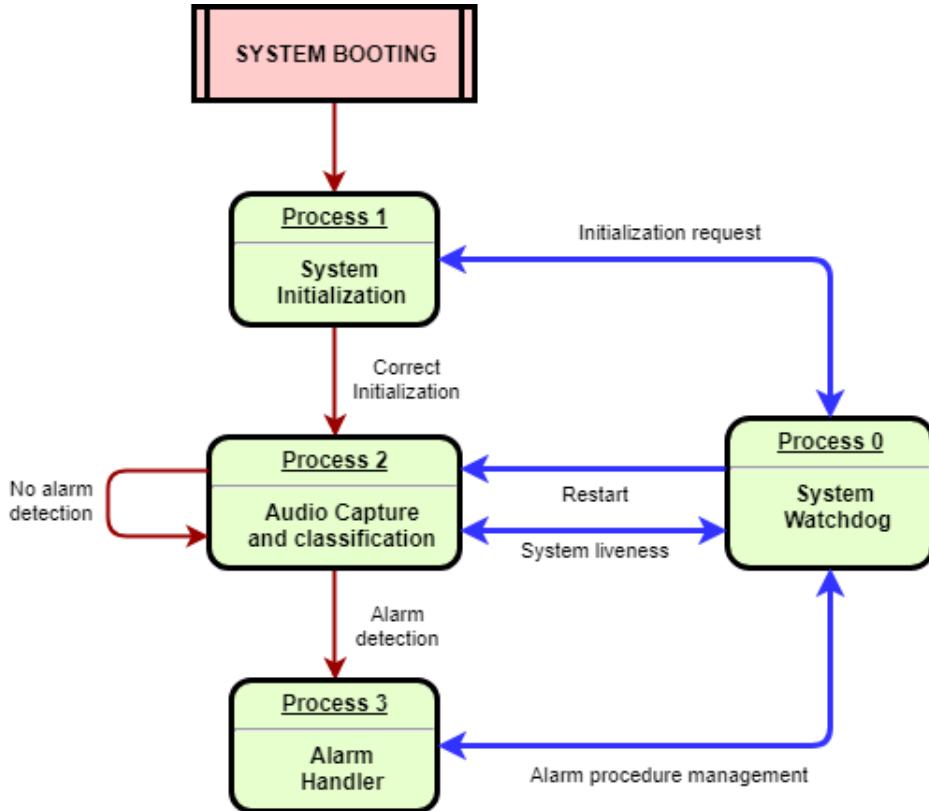
**Figure 4.29:** Ranger Vervet complete wiring diagram. This diagram unites all the previous schemas, showing the entire system wiring. The MCU can be supplied with a USB charger.

# **Chapter 5**

## **Software Specifications**

## 5.1 Software Architecture

The core of Ranger Vervet is a multi-module software that operates in order to complete a group of processes whose interaction determines the flowchart describing the system behavior, as in figure 5.1.

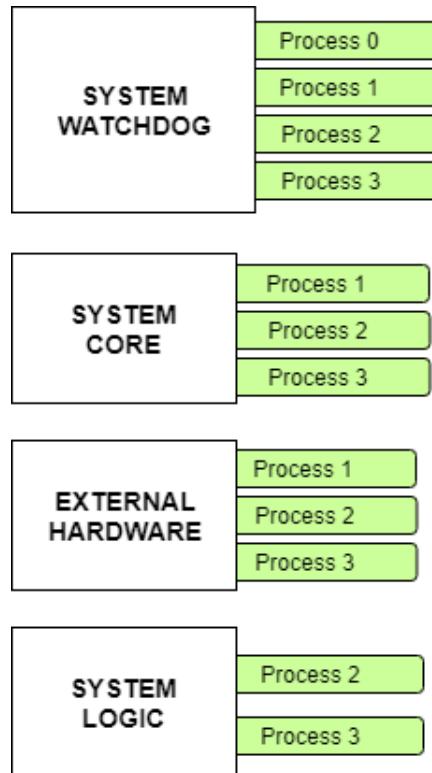


**Figure 5.1:** Ranger Vervet software flowchart. In green the 4 main processes of the software, red arrows indicate the events that make the flow move from one process to the other. Blue arrows are special in the sense that represent control events that have to be managed only by the System Watchdog process to guarantee a correct system functioning.

From a high level point of view, the software architecture that implements the previous flowchart is made up of only 4 elements, each of which contains the single software modules that build the entire system. Figure 5.2 shows these macro-blocks and the relationship with the processes mentioned in the software flowchart. The main role of each macro block is stated below.

- **System Watchdog:** integrity checker and components coordinator;
- **System Core:** driver software of the MCU, including the drivers to control its native peripherals;

- **External Hardware:** comprises all the drivers for external devices attached to the MCU;
- **System Logic:** component dedicated to the audio analysis and signal classification and to the handling of the alarm detected.



**Figure 5.2:** Ranger Vervet software macro-modules.

### 5.1.1 Modules Details

Each of the four previously introduced modules is made up of header and source files written in C language using the STMicroelectronics HAL Library to interface with the processor. Almost all blocks contain couples of header and source files. The next notes show the available interfaces to the user for each source file belonging to the four macro-blocks. Well note that this does not hold for the System Core source code which is property of STMicroelectronics and ARM, and whose usage can be investigated referring to the official guides for their own libraries [19].

## System Watchdog

Tree composition and description:

System Watchdog

```
| - SystemWatchdogInc
| | system_watchdog.h
| - SystemWatchdogSrc
| | system_watchdog.c
```

Prototype	Description
bool system_init()	Initializes the Ranger Vervet system, device by device. <i>Param:</i> none <i>Return:</i> true if the system is correctly initialized, false otherwise
bool check_system_status()	Checks the dynamic parameters of the Ranger Vervet system, such as supply voltage level or GSM signal, to verify that is correctly working. <i>Param:</i> none <i>Return:</i> true if the device is correctly working, false otherwise
void alarm_timeout()	Controls the alarm handling procedure and stops it according to the detected alarm. <i>Param:</i> none <i>Return:</i> none
void system_shutdown()	Shuts down the Ranger Vervet system by deinitializing all the processor peripherals and putting the camera and GSM board in sleep mode and the processor in standby. <i>Param:</i> none <i>Return:</i> none

**Table 5.1:** system\_watchdog.h interface

## System Core

This module is written using STMicroelectronics HAL Library and STM CubeMX wizard. For explicit details refer to [19] [20].

Tree composition and description:

System Core

```
| - SystemCoreInc
| | adc.h
```

```
| | dma.h  
| | gpio.h  
| | i2c.h  
| | tim.h  
| | usart.h  
| | stm32f4xx_it.h  
| | stm32f4xx_hal_conf.h  
| | main.h  
| | common.h  
|- SystemCoreSrc  
| | adc.c  
| | dma.c  
| | gpio.c  
| | i2c.c  
| | tim.c  
| | usart.c  
| | stm32f4xx_it.c  
| | system_stm32f4xx.c  
| | stm32f4xx_hal_msp.c
```

A special mention goes to `common.h`, which is the interface that declares all the C `extern` variables with global scope that are fundamental to connect and synchronize all the source modules.

## External Hardware

Tree composition and description:

External Hardware

```
|- ExternalHardwareInc  
| | camera.h  
| | GSM_board.h  
| | led_matrix.h  
| | microphone.h  
| | motor_driver.h
```

```

| | OV7670_register_definitions.h
| | speaker_alarm.h
| - ExternalHardwareSrc
| | camera.c
| | GSM_board.c
| | led_matrix.c
| | microphone.c
| | motor_driver.c
| | speaker_alarm.c

```

Prototype	Description
void camera_init()	Initializes the OV7670 camera module setting its registers. <i>Param:</i> none <i>Return:</i> none
bool camera_check()	Checks if camera registers are correctly set. <i>Param:</i> none <i>Return:</i> none
void camera_mode(bool mode)	Sets the camera operating mode. <i>Param:</i> mode true for normal mode, false for sleep mode <i>Return:</i> none
void camera_capture_image()	Captures an image and sends it via UART. <i>Param:</i> none <i>Return:</i> none

**Table 5.2:** camera.h interface

Prototype	Description
turn_on_led_matrix()	Lights up the LED matrix. <i>Param:</i> none <i>Return:</i> none
turn_off_led_matrix()	Turns off the LED matrix. <i>Param:</i> none <i>Return:</i> none
init_led_matrix()	Initializes the LED matrix. <i>Param:</i> none <i>Return:</i> none

**Table 5.3:** led\_matrix.h interface

Prototype	Description
bool GSM_init()	<p>Initializes the GSM board connecting to the network.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> true if GSM module is correctly initialized, false otherwise</p>
bool GSM_check()	<p>Verifies if the module is in good power and signal status to properly work.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> true if the GSM board is correctly working, false otherwise</p>
bool GSM_send_HTTP()	<p>Sends HTTP POST command to a remote server.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> true if HTTP command is correctly executed, false otherwise</p>
bool GSM_mode(bool mode)	<p>Sets the GSM board in sleep mode or active mode.</p> <p><i>Param:</i> mode true if active mode, false otherwise</p> <p><i>Return:</i> true if command is correctly executed, false otherwise</p>
bool GSM_GPRS_init()	<p>Initializes GPRS parameters for Internet connection.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> true if GPRS is correctly initialized, false otherwise</p>
bool GSM_send_SMS()	<p>Sends SMS to a receiver terminal.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> true if SMS is correctly sent, false otherwise</p>

**Table 5.4:** `GSM_board.h` interface

Prototype	Description
bool microphone_check()	<p>Checks if microphones work properly.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> If true microphones work correctly otherwise returns false</p>
void record_audio()	<p>Records the audio ADC converted sensed by microphones.</p> <p><i>Param:</i> none</p> <p><i>Return:</i> none</p>
void microphone_mode(bool mode)	<p>Enables or disables audio recording.</p> <p><i>Param:</i> mode If true turns on microphones else turns off microphones</p> <p><i>Return:</i> none</p>

**Table 5.5:** `microphone.h` interface

Prototype	Description
motor_turn(uint8_t direction)	Turns the motor of 45 degrees from center to left or from center to right depending on the alarm localization. <i>Param:</i> integer indicating left or right <i>Return:</i> none
motor_default(uint8_t current_position)	Brings the motor to the steady central position. <i>Param:</i> current_position integer indicating the current position, left or right from the center <i>Return:</i> none
init_motor(bool* end_of_run)	Sets the home position of the motor. <i>Param:</i> end_of_run pointer to a flag indicating the movement line checking <i>Return:</i> none

**Table 5.6:** motor\_driver.h interface

Prototype	Description
void start_alarm_sound(uint16_t freq)	Given an assigned frequency, generates an alarm sound at that frequency. <i>Param:</i> freq Integer representing the alarm frequency htim timer handler managing the PWM generation <i>Return:</i> none
void stop_alarm_sound()	Stops the sound generation. <i>Param:</i> none <i>Return:</i> none

**Table 5.7:** speaker\_alarm.h interface

## System Logic

Part of this software, in particular the files arm\_math.h, arm\_fir\_init\_f32.c, arm\_fir\_f32.c, are property of ARM Keil and belong to the CMSIS DSP Library and will not be studied in depth. To get more details refer to [21].

Tree composition and description:

System Logic

```

|- SystemLogicInc
| | arm_math.h
| | alarm_handler.h
| | classifier.h
| - SystemLogicSrc
| | alarm_handler.c

```

```
| | classifier.c
| | arm_fir_init_f32.c
| | arm_fir_f32.c
```

Prototype	Description
bool handle_alarm(char alarm_type)	<p>According to the alarm signaled handles it with the corresponding procedure.</p> <p><i>Param:</i> alarm_type char indicating the alarm type  <i>Return:</i> none</p>

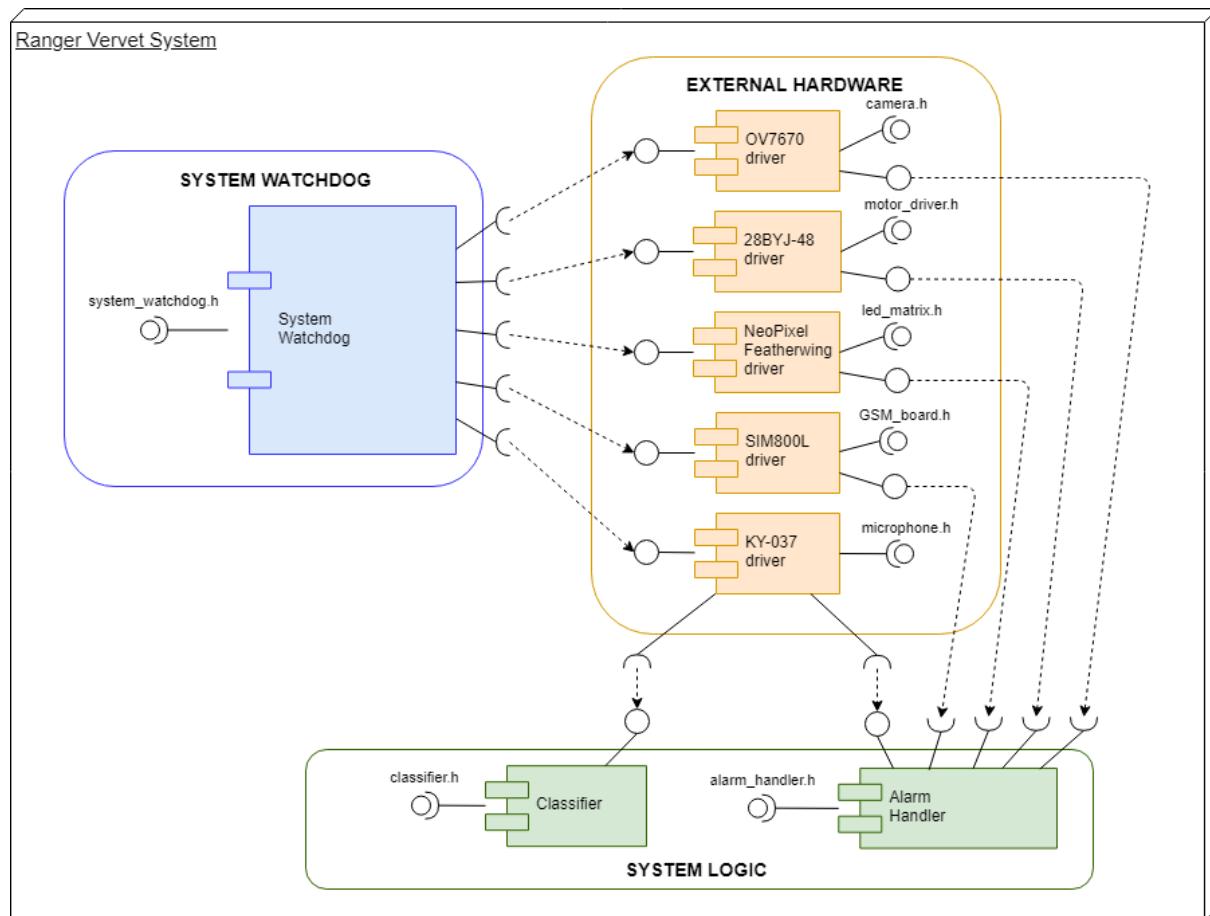
**Table 5.8:** alarm\_handler.h interface

Prototype	Description
char classify(float32_t* input)	<p>Classifies the input signal according to the 4 alarm classes.</p> <p><i>Param:</i> input points to input signal sequence  <i>Return:</i> identifier of the input class recognized by the algorithm, 'e' for aerial alarm, 'l' for terrestrial alarm, 's' for snake alarm, 'n' for no alarm</p>

**Table 5.9:** classifier.h interface

## Software architecture component diagram

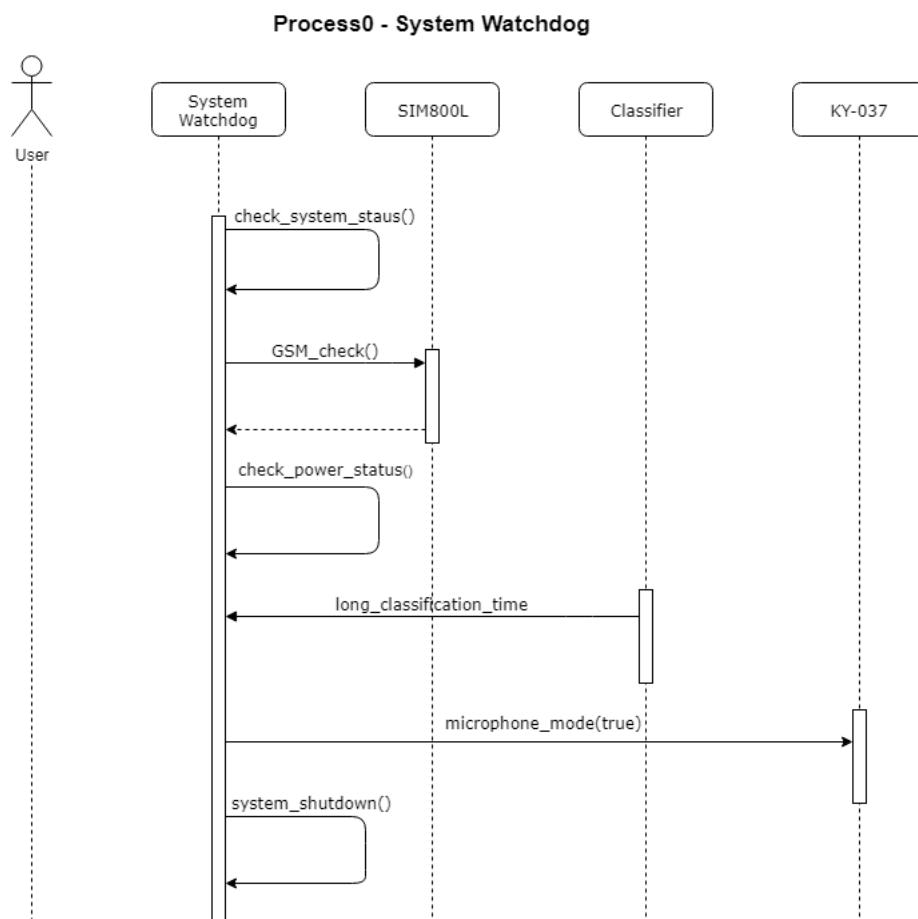
By knowing deeper details about the macro-modules of Ranger Vervet System software, is now possible to represent with major details the comprehensive software infrastructure, available in figure 5.3.



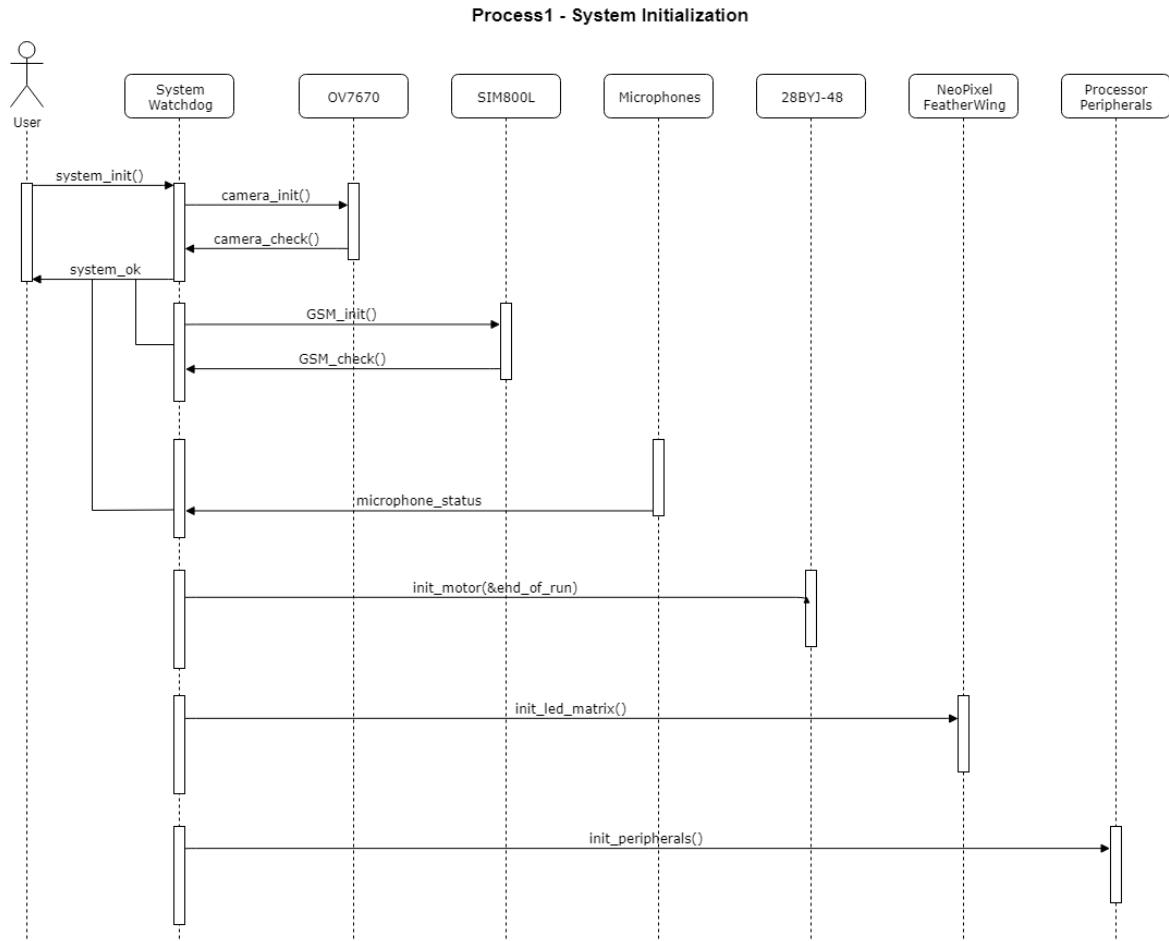
**Figure 5.3:** Ranger Vervet System component diagram. All required interfaces correspond to the provided interfaces by a given module. For simplicity this diagram does not make any reference to the System Core block.

## 5.2 Software dynamic behavior

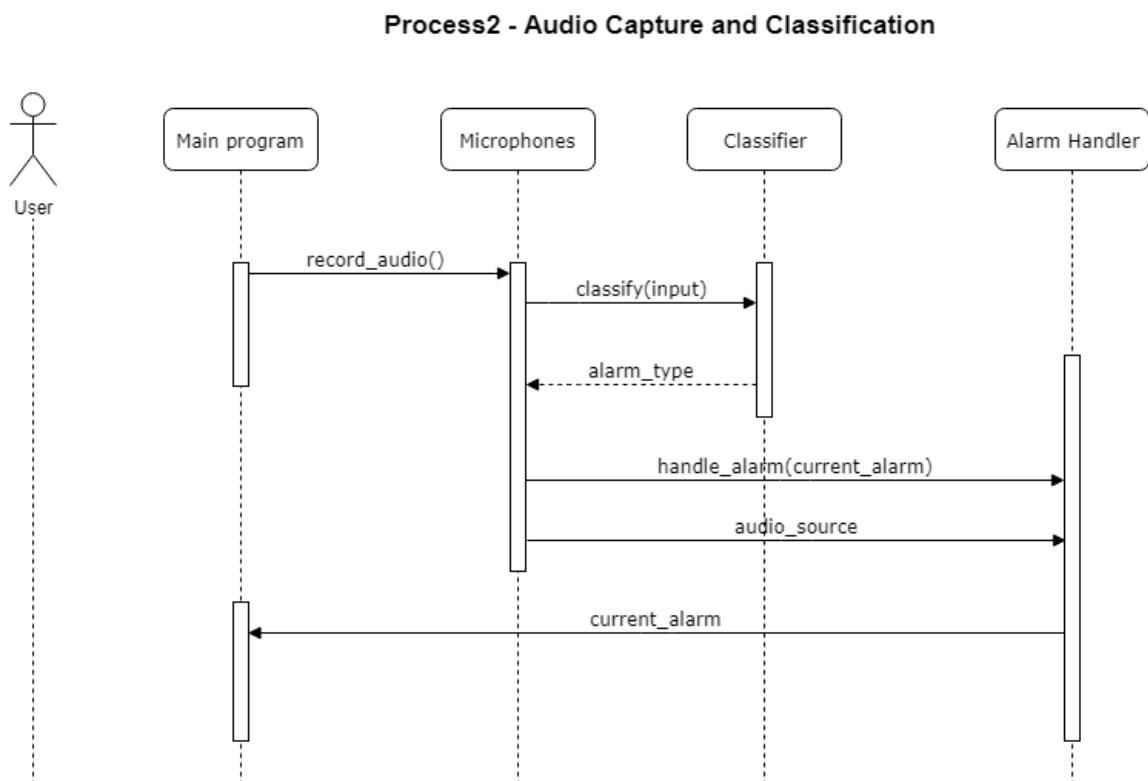
The high level view of the flowchart crossed by Ranger Vervet software is now expanded, showing a sequence diagram to explain the processes cited in 5.1, taking as reference the modules details of the previous section.



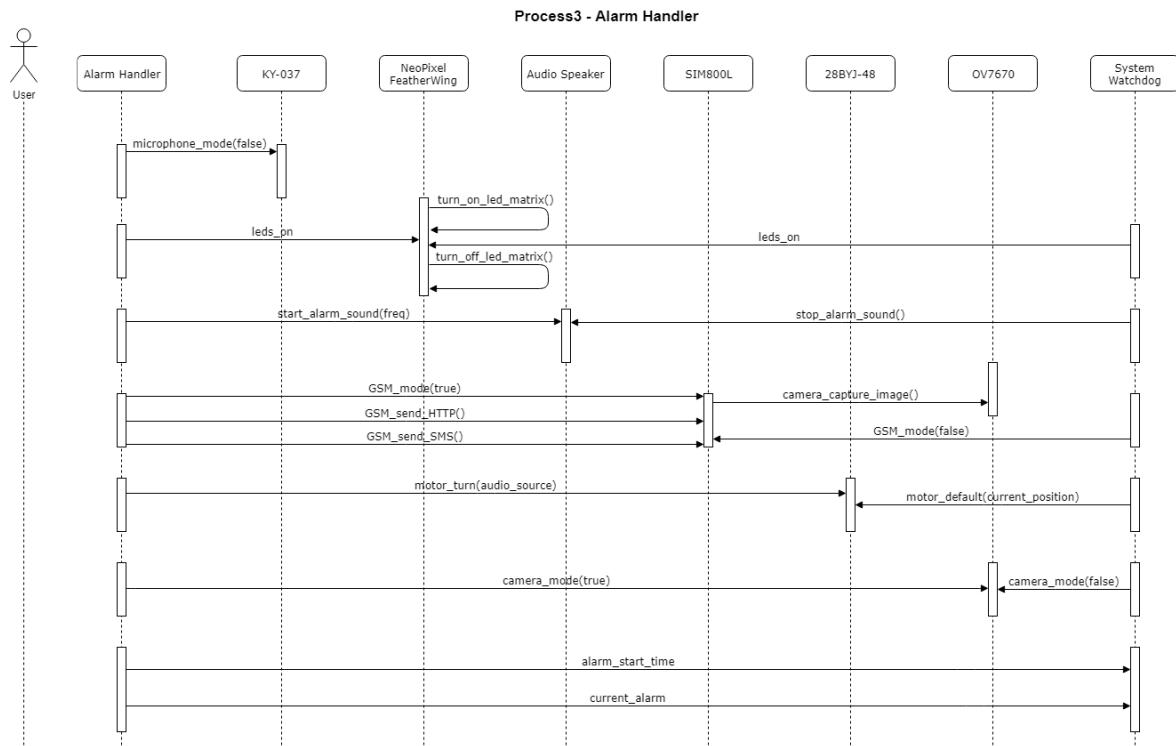
**Figure 5.4:** System Watchdog sequence diagram. Periodically System Watchdog retrieves the fundamental information about the system state, which are the network connection status of SIM800L and its battery charge level, voltage supply of the processor, and rate of convergence of the classification algorithm. If any of these data is faulty, Ranger Vervet is immediately shutdown. Moreover System Watchdog resets the procedure of audio listening that can be suspended when any alarm is handled.



**Figure 5.5:** System Initialization sequence diagram. When the system is booted, System Watchdog queries all devices for initialization, it receives an acknowledgement from some of them, in form of return of function call or modification of globally scoped variables. If any of them is faulty according to its response, Ranger Vervet cannot boot and shuts down. In this process also all processor peripherals are initialized using System Core routines provided by STM. The `init_peripherals()` is a placeholder for the set of specific function that manage to start the involved MCU peripherals (timers, DMA, serial connections ecc.). In this case the process starts by external input, where the actor is a common user that physically turns on Ranger Vervet.



**Figure 5.6:** Audio Capture and Classification sequence diagram. In the main program an asynchronous call to `record_audio()` builds sample by sample the audio chunk to analyze passed as argument to the Classifier. After the digital signal processing is completed, Classifier returns the type of the alarm detected, if one exists. Then a call to the Alarm Handler is performed, the alarm handling procedure starts on the basis of the current alarm detected by Classifier. Alarm Handler also sets a global variable, `current_alarm`, that informs the system about the current state. This variable is periodically scanned by System Watchdog. The variable `audio_source` indicates the direction of the incoming alarm call, necessary in case of terrestrial threat.



**Figure 5.7:** Alarm Handler sequence diagram. Depending on the type of the alarm, Alarm Handler starts a procedure by sending activation commands to the needed devices. Nevertheless, the first step is sending to System Watchdog a synchronization information, `alarm_start_time`, about the exact system time of start of alarm handling and then stopping audio recording with `microphone_mode(false)`. By means of this, System Watchdog can initiate a timeout counting to stop the handling procedure and no CPU time is wasted in recording unusable data. When the timeout expires System Watchdog restores Ranger Vervet initial condition, essentially resetting the devices participating to the alarm handling routine. It is noticeable how SIM800L asks directly to OV7670 to get the image to transmit without passing for other modules.

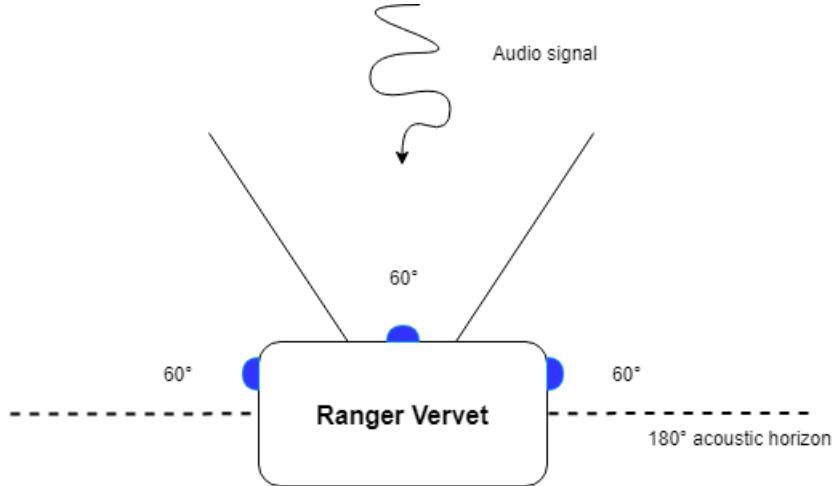
## 5.3 Ranger Vervet Skills

The next sections are dedicated to a brief, but necessary, explanation of the main algorithms that lie behind some of the Ranger Vervet System behavior. There are essentially two aspects to investigate, and that allow to satisfy Func-Requirements 5,7,8, which are audio localization and audio classification.

### 5.3.1 Audio Localization

Many sophisticated algorithms are able to localize a signal source, such as Angle-of-Arrival Estimation (AOA), Time-Difference-of-Arrival Estimation (TDOA), MULTiple-Signal-Classification (MUSIC) and so on. All of these require high quality hardware to capture the signal, and work in very specific conditions. That's why Ranger Vervet uses a simpler but effective approach based on input signal energy to approximate the angle of arrival of the audio. Given an acoustic horizon of 180 degrees, three microphones allow to split it with a resolution of 60 degrees. This means that for Ranger Vervet only three possible directions are known, as shown in 5.8.

Ideally, each of the three microphones records an input sequence made of 5000 samples at



**Figure 5.8:** Audio localization representation. Blue elements portray microphones, which divide the input audio angle of arrival in three sections.

10kHz, namely  $x_1[n], x_2[n], x_3[n]$  s.t.  $n \in [0, 4999]$ . These signals are characterized by a finite energy, which is the Euclidean norm given by

$$E_x = \sum_{n=0}^{4999} x^2[n] \quad (5.1)$$

Ranger Vervet builds iteratively this value for each of the three microphones at every captured sample, in such a way to evaluate the energy of the audio signal at each channel. The maximum energy associated to a microphone indicates with good probability the direction of the incoming audio, with better saying the angular section from which the sound wave is arriving. So the decision rule implemented to localize the audio source is the following

$$ArrSection = \operatorname{argmax}_{i \in \{1,2,3\}} E_i \quad (5.2)$$

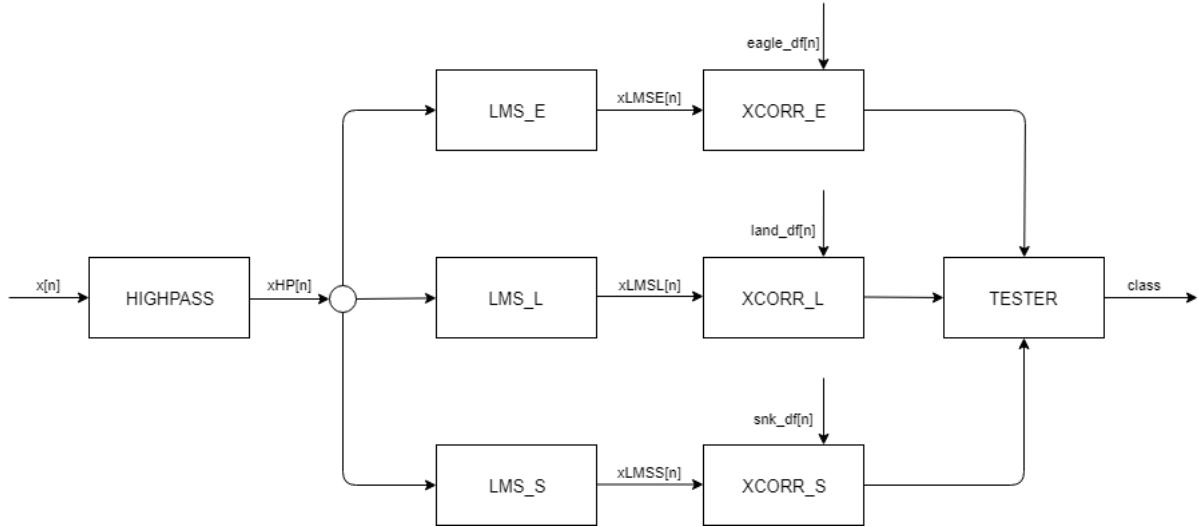
where  $E_i$  is defined as in 5.1 and with the subsequent angular mapping for the AOA given by  $ArrSection = 1 \Rightarrow AOA \in [180, 120]$ ,  $ArrSection = 2 \Rightarrow AOA \in [120, 60]$ ,  $ArrSection = 3 \Rightarrow AOA \in (60, 0]$ .

### 5.3.2 Recorded audio signal construction

At each ADC conversion cycle completion, Ranger Vervet has three different audio samples, corresponding to the three KY-037 microphones. To decide which of them is the sample to save in audio chunk that is currently recording, Ranger Vervet exploits again equation 5.1. It stores as freshly recorded sample the one corresponding to the input sequence with higher partial energy.

### 5.3.3 Audio Classification

The algorithm used to classify the input signal sequence is based on adaptive filtering theory, in particular on LMS (Least Mean Squares) filtering. An LMS filter is a FIR filter whose aim is to minimize the MSE during an estimation problem. Given an input signal an LMS filter processes it to estimate a given signal with the smallest error possible. For this reason it is a very simple technique of supervised learning. Under the hood it requires an algorithm that autonomously changes the filter weights in to minimize the square error, and into specific it is a stochastic steepest descent algorithm. Refer to [22] for further information. The block diagram in 5.9 shows the dataflow of the classification algorithm executed by Ranger Vervet. The input signal, let us assume  $x[n]$ , is the sequence built up sample by sample recording the environmental audio according to 5.3.2. The first step consists in reducing any low frequency noise that can corrupt the results. This is accomplished with digital high pass filtering of the signal, as also suggested by [1], with stopband at 0.1KHz. The highpass filter has been designed with the



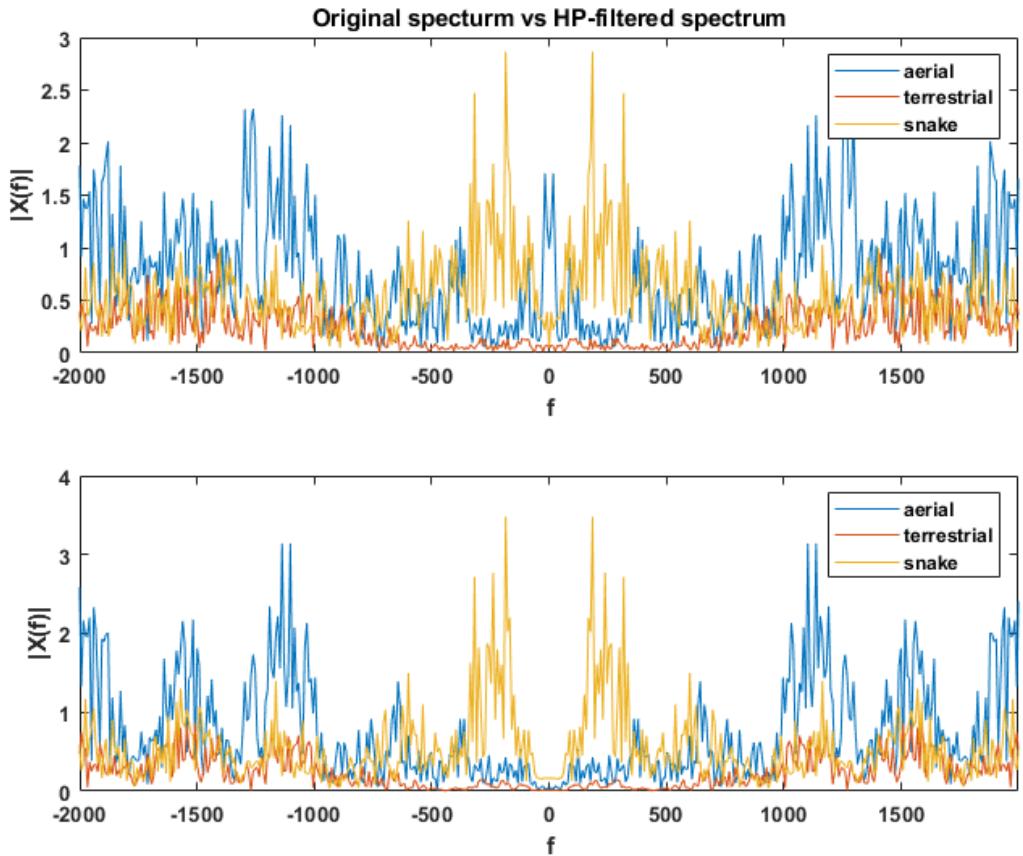
**Figure 5.9:** Classification algorithm data flow. Each block is implemented with typical DSP operations and filters.

MATLAB Filter Designer Toolbox using the equiripple optimal approximation. The result is the new signal  $xHP[n]$ . Figure 5.10 shows an example of this process.

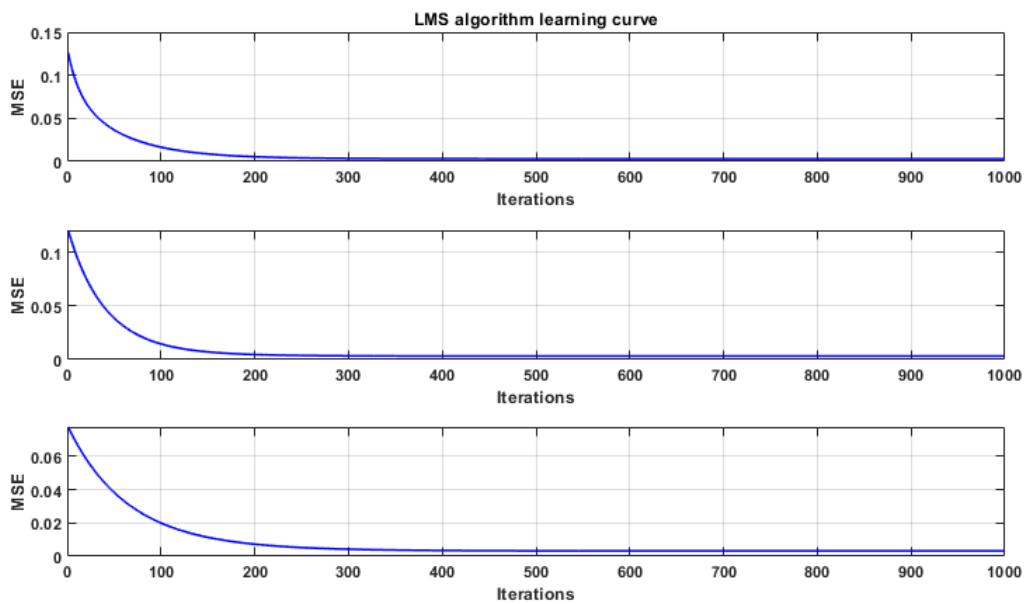
$xHP[n]$  is newly filtered, but this time using as FIR filter the tap weights obtained using the LMS algorithm. These three filters, in figure LMS\_E, LMS\_L, LMS\_S, have been obtained by preliminarily training an LMS adaptive filter for each alarm call category, in order to minimize the estimation error of the input signal with respect to a training set for each category. Doing so, Ranger Vervet makes a best effort estimation for each alarm category, and produces three output signals,  $xLMSE[n]$ ,  $xLMSL[n]$ ,  $xLMSS[n]$ .

The last three sequences are cross-correlated with the corresponding training signal of the alarm category. In this way a similitude indicator is produced. Well note that Ranger Vervet calculates a sampled correlation function for the inherent computational complexity of the function.

Finally, a threshold mechanism checks which is the maximum correlation value observed for each category. If the highest correlation doubles at least the second highest correlation value then the classifier notifies an alarm for this category, otherwise it states that no alarm is detected.



**Figure 5.10:** Highpass filtering of alarm calls. It is evident how low frequency signal components have been reduced in amplitude after filtering



**Figure 5.11:** LMS algorithm learning curve. This three plots show the MSE trend for the three alarm calls against the number of iterations of the LMS algorithm. It is clear how the algorithm converges to a finite error, around 0.01. From up to down: aerial alarm call, terrestrial alarm call, snake alarm call. These data come from MonteCarlo simulations on MATLAB.

# References

- [1] Tabitha Price et al. "Vervets revisited: A quantitative analysis of alarm call structure and context specificity". In: *Scientific reports* 5 (2015), p. 13220.
- [2] Michael J Owren and Robert H Bernacki. "The acoustic features of vervet monkey alarm calls". In: *The Journal of the Acoustical Society of America* 83.5 (1988), pp. 1927–1935.
- [3] *STM32F401xB/C and STM32F401xD/E advanced Arm®-based 32-bit MCUs*. RM0368. Rev. 5. STMicroelectronics. Dec. 2018.
- [4] *STM32 Nucleo-64 boards*. UM1724. Rev. 12. STMicroelectronics. Dec. 2017.
- [5] *Arm® Cortex®-M4 32b MCU+FPU, 105 DMIPS, 256KB Flash/64KB RAM, 11 TIMs, 1 ADC, 11 comm. interfaces*. 024738. Rev. 10. STMicroelectronics. Dec. 2017.
- [6] *SIM800 Hardware Design*. Rev. 0. SIMTech. Aug. 2013.
- [7] *SIM800 Series AT Command Manual*. Rev. 9. SIMTech. Aug. 2015.
- [8] *SIM800 Series\_IP\_Application Note*. Rev. 0. SIMTech. Aug. 2013.
- [9] *OV7670/OV7171 CMOS VGA CameraCHip with OmniPixel® Technology Advanced Information Preliminary Datasheet*. Rev. 1. OmniVision Technologies. July 2005.
- [10] *OV7670/OV7171 CMOS VGA CameraCHip Implementation Guide*. Rev. 0. OmniVision Technologies. Sept. 2005.
- [11] *OV7670 Software Application Note*. OmniVision Technologies.
- [12] *OmniVision Serial Camera Control Bus (SCCB) Functional Specification*. Rev. 4. OmniVision Technologies. July 2007.
- [13] *Sound Detector Hookup Guide*. sparkfun Technologies. URL: <https://learn.sparkfun.com/tutorials/sound-detector-hookup-guide>.
- [14] *WS2812 Intelligent control LED integrated light source*. WorldSemi.

- [15] *Adafruit NeoPixel FeatherWing*. Adafruit Industries. Nov. 2018.
- [16] *LM386 Low Voltage Audio Power Amplifier*. SNAS545C. Rev. 3. Texas Instruments. May 2017.
- [17] *28BYJ-48 – 5V Stepper Motor*. Kiatronics.
- [18] *ULN2001A-ULN2002A ULN2003A-ULN2004A Seven Darlington Arrays*. STMicroelectronics. Feb. 2002.
- [19] *Description of STM32F4 HAL and LL drivers*. UM1725. Rev. 5. STMicroelectronics. Feb. 2017.
- [20] *STM32CubeMX for STM32 configuration and initialization C code generation*. UM1718. Rev. 29. STMicroelectronics. Apr. 2019.
- [21] *CMSIS DSP Software Library*. Keil. URL: <http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>.
- [22] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2005.
- [23] *Bonus Feature - Alcoholic Vervet Monkeys! - Weird Nature - BBC animals*. BBC. URL: <https://www.youtube.com/watch?v=pSm7BcQHWXk>.

# List of Figures

2.1	Friendly Vervet Monkeys.	4
3.1	UML Use Case Diagram for Requirement 1.	6
3.2	UML Use Case Diagram for Requirement 6.	7
3.3	UML Use Case Diagram for Requirement 8.	8
4.1	Block level hardware architecture.	11
4.2	STMF401RE board.	12
4.3	STMF401RE circuit diagram.	15
4.4	SIMCOM 800L GSM Module.	15
4.5	SIMCOM800L GSM Module functional diagram.	16
4.6	SIMCOM 800L power supply diagram	17
4.7	SIMCOM800L GSM Module serial connection.	17
4.8	OmniVision OV7670 Camera Module.	18
4.9	OmniVision OV7670 Camera Module hardware architecture.	19
4.10	OV7670 camera module timing diagram	21
4.11	KY-037 Microphone Sensor Module	22
4.12	KY-037 sound detector functional block diagram.	23
4.13	NeoPixel ws2812 pin diagram.	23
4.14	NeoPixel ws2812 pin diagram.	24
4.15	ws2812 commands timing diagram.	24
4.16	ws2812 daisy chaining.	25
4.17	Ranger Vervet audio speaker	26
4.18	LM386 audio amplifier.	26
4.19	Audio Speaker application circuit	27
4.20	28BYJ-48 Stepper Motor.	27

LIST OF FIGURES	56
4.21 28BYJ-48 pinout representation using internal motor coils. . . . .	28
4.22 ULN2003 stepper motor driver. . . . .	29
4.23 SIMCOM800L - STM32F401RE connection . . . . .	30
4.24 OV7670 - STM32F401RE connection . . . . .	31
4.25 KY037 - STM32F401RE connection . . . . .	31
4.26 Adafruit NeoPixel FeatherWing - STM32F401RE connection . . . . .	32
4.27 Audio Speaker - STM32F401RE connection . . . . .	32
4.28 28BYJ-48 - STM32F401RE connection . . . . .	33
4.29 Ranger Vervet complete wiring diagram. . . . .	34
 5.1 Ranger Vervet software flowchart. . . . .	36
5.2 Ranger Vervet software macro-modules. . . . .	37
5.3 Ranger Vervet System component diagram. . . . .	44
5.4 System Watchdog sequence diagram. . . . .	45
5.5 System Initialization sequence diagram. . . . .	46
5.6 Audio Capture and Classification sequence diagram. . . . .	47
5.7 Alarm Handler sequence diagram. . . . .	48
5.8 Audio localization representation . . . . .	49
5.9 Classification algorithm data flow . . . . .	51
5.10 Highpass filtering of alarm calls. . . . .	52
5.11 LMS algorithm learning curve. . . . .	52

# List of Tables

4.1	SIMCOM 800L GSM Module pinout . . . . .	16
4.2	SIMCOM 800L GSM serial interface voltage ratings. . . . .	17
4.3	OV7670 Camera Module pinout. . . . .	20
4.4	KY-037 sound detector pinout. . . . .	22
4.5	NeoPixel ws2812 pinout. . . . .	24
4.6	NeoPixel ws2812 timing specifications. . . . .	24
4.7	28BYJ-48 Stepper Motor pinout. . . . .	28
4.8	28BYJ-48 Stepper Motor switching sequence. . . . .	28
4.9	ULN2003 motor driver pinout. . . . .	29
5.1	system_watchdog.h interface . . . . .	38
5.2	camera.h interface . . . . .	40
5.3	led_matrix.h interface . . . . .	40
5.4	GSM_board.h interface . . . . .	41
5.5	microphone.h interface . . . . .	41
5.6	motor_driver.h interface . . . . .	42
5.7	speaker_alarm.h interface . . . . .	42
5.8	alarm_handler.h interface . . . . .	43
5.9	classifier.h interface . . . . .	43