MOBILE MAPPING CAR

HW Architecture & Design

Supervisor: MSc. Torben Gregersen

Team: AAD1

Editors:

- Guillaume Trousseau
- Jaime Escuer
- Iñaki Abadia
- Roger Prat

AARHUS UNIVERSITY

Applied App Development
(ITIPRJ)

Content

1 Hardware Architecture

1.1 Introduction:

This document is based on the requirements specification document.

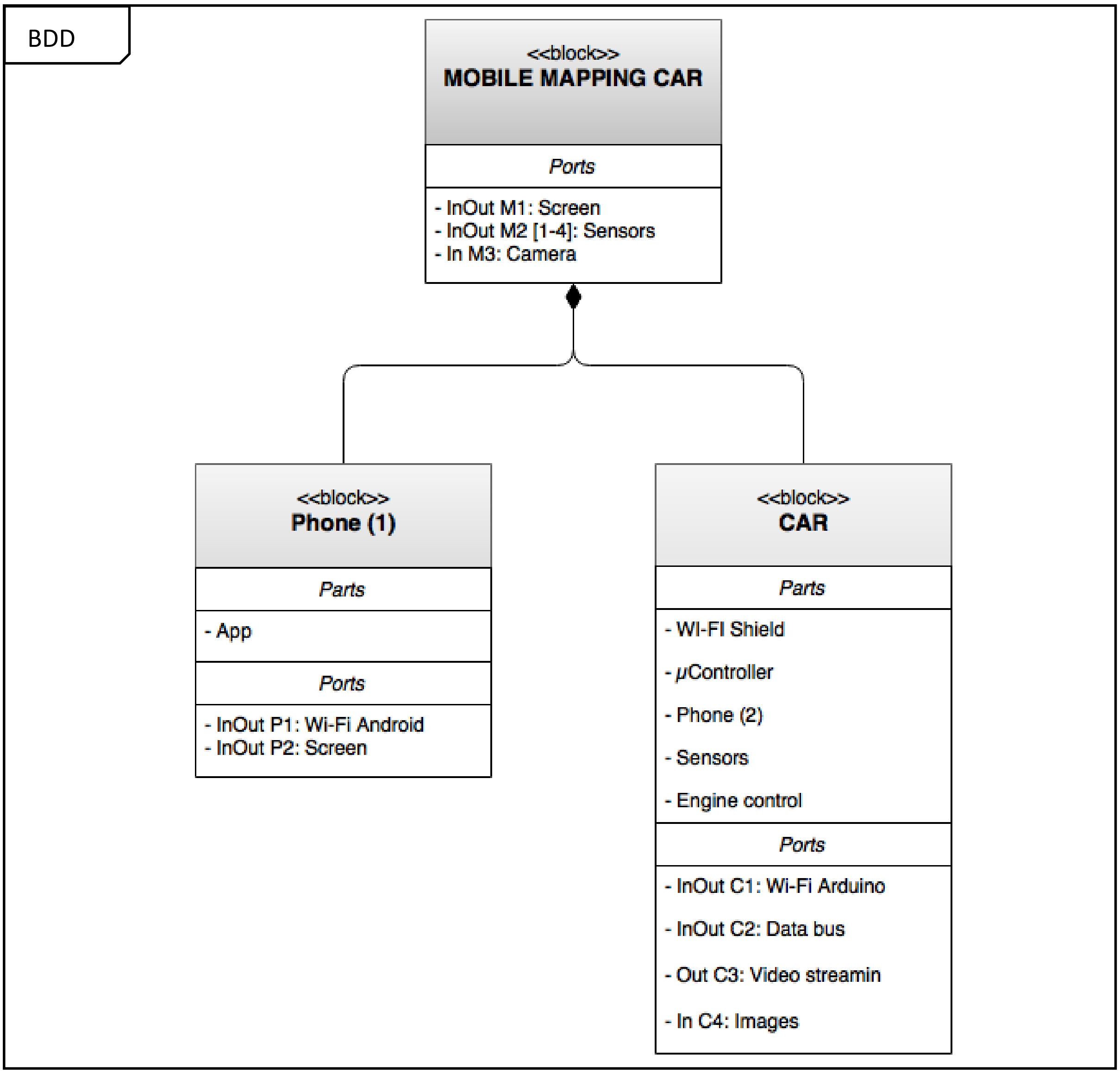
We explain the Hardware architecture and design of our system.

To decompose the system and explain it, we use SysML Language including BDD and IBD where we explain the internal and external connections of the system.

1.2 BDD Mobile Mapping Car

The next figure shows the main system blocks and their connections.

The “Phone (1)” controls the car. We have to emphasize that we do the communication between devices via Wi-Fi.



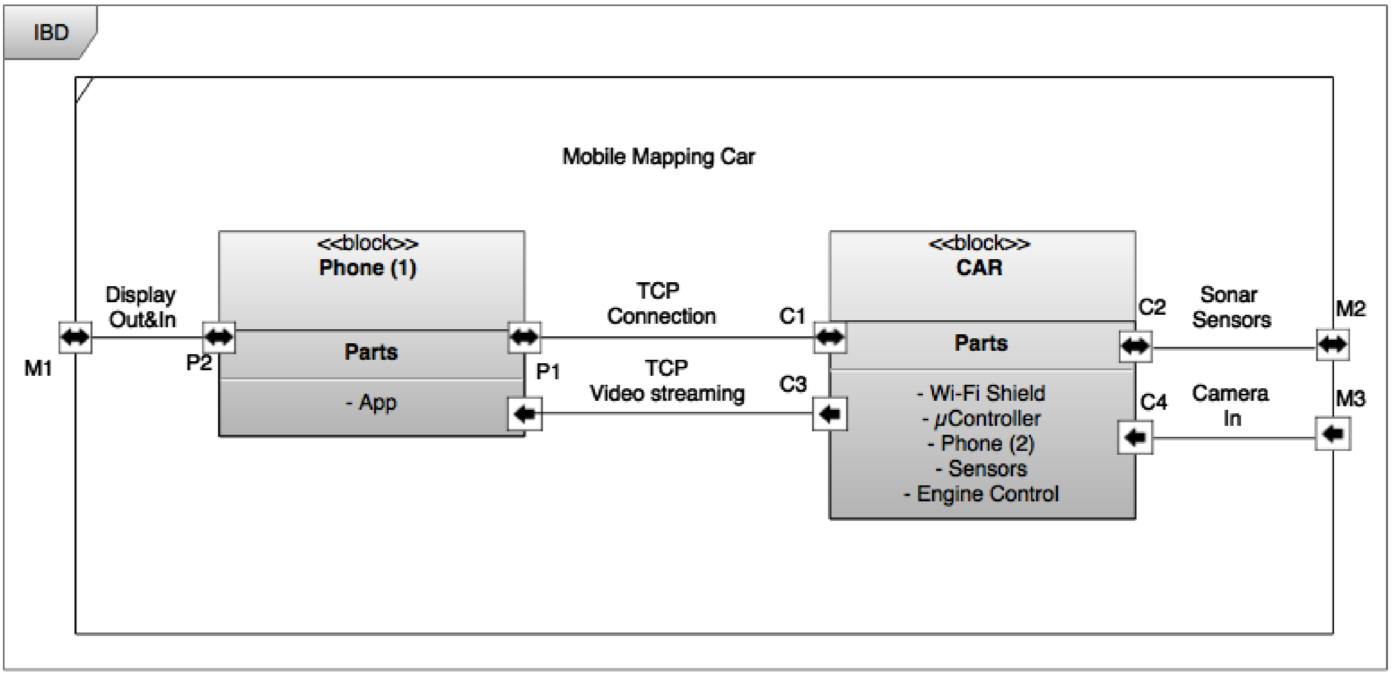
1.2.1 Explicative table BDD MMC

Block name	***Function Description***	***Port Name***	***Direction***	***Comment***
Mobile Mapping Car	This block shows the integration of the devices that make our system.	M1	In/Out	User interaction with smartphone
		M2 [1-6]	In/Out	Port to send and receive signals from the sensors
		M3	In	Images obtained with the camera
Phone (1)	This Smartphone has the App to control the car movements and it also starts recording maps mode.	P1	In/Out	Through Wi-Fi connection we can send messages to the car and receive data from the Arduino board
		P2	In/Out	The user controls the system through the screen
Car	It is the main section, It consists in a microcontroller controlling all the sensors and communicating with the main Smartphone.	C1	In/Out	The Wi-Fi shield creates a communication between the Car and the “Phone(1)”
		C2	In/Out	We connect the sensors and the microcontroller. With this connections we activate the sensors and we also receive the data form them
		C3	Out	We have a second Phone where we have an App for video streaming. This phone sends the images to the Phone(1) by Wi-Fi
		C4	In	Images obtained by the camera

1.3 IBD Mobile Mapping Car

This diagram shows the different ways of communication, inside and outside of our system.

The arrows indicate the direction of each port for the communication between the blocks.

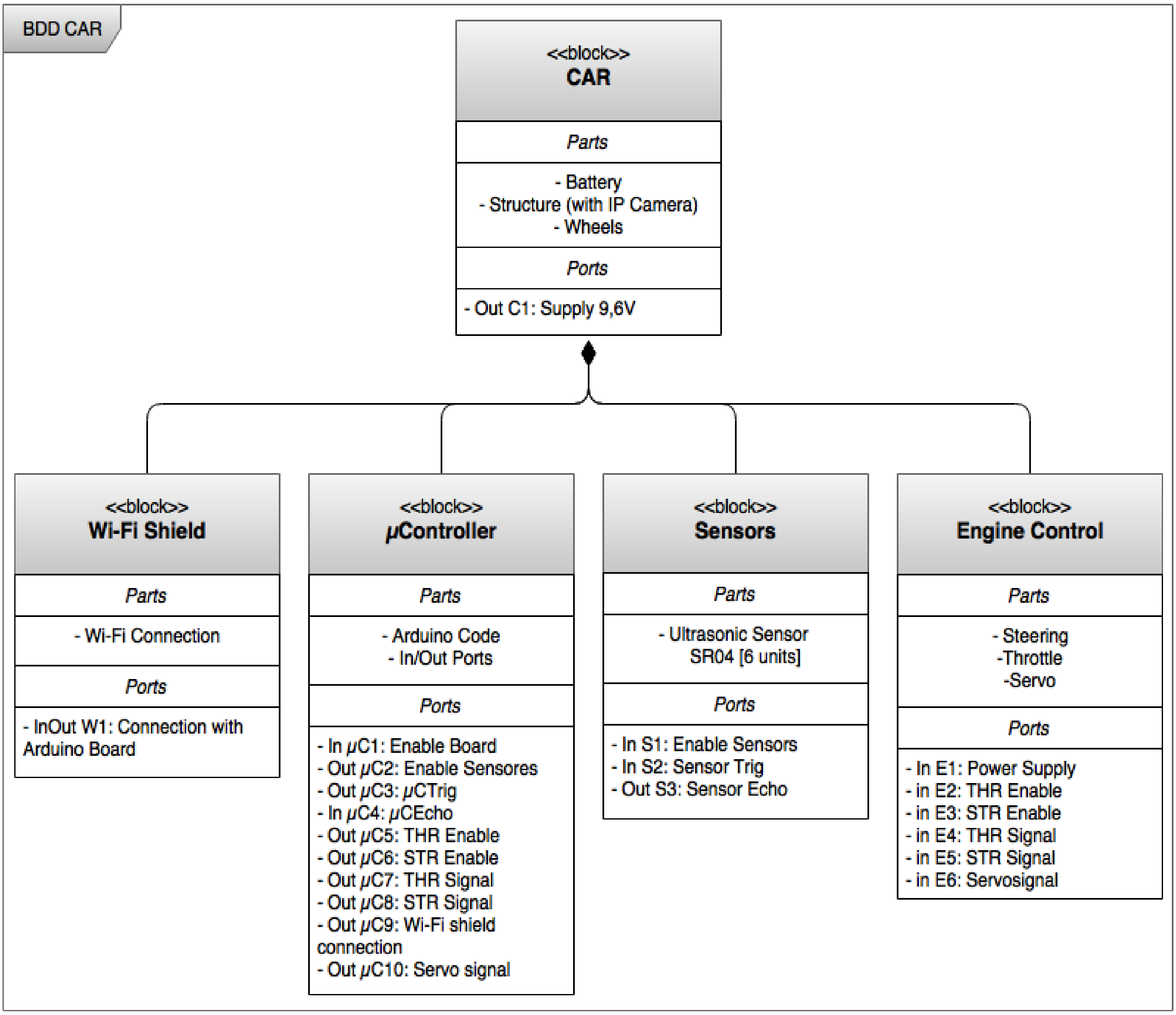


1.3.1 Explicative table IBD MMC

Name	*Description*	*Port 1*	*Port 2*
Display	User interaction with the main App.	M1	P2
TCP Connection Android - Arduino	Creates the socket between the main Smartphone and the Arduino board to send and receive messages and instructions.	P1	C1
TCP Video streaming	The Phone (2) sends video streaming to Phone (1) via TCP protocol.	P1	C3
Sonar Sensors	In this connection we have some cables to enable the four sensors. We also have "Trig" and "Echo" connections to enable measurement and obtain data.	C2	M2
Camera In	Phone 2 records the live video using an application to run as an IP camera.	C4	M3

1.4 BDD CAR

This diagram shows in more detail the Car block. Here we have all the hardware parts and we can see how these components are connected together.



1.4.1 Explicative table BDD CAR

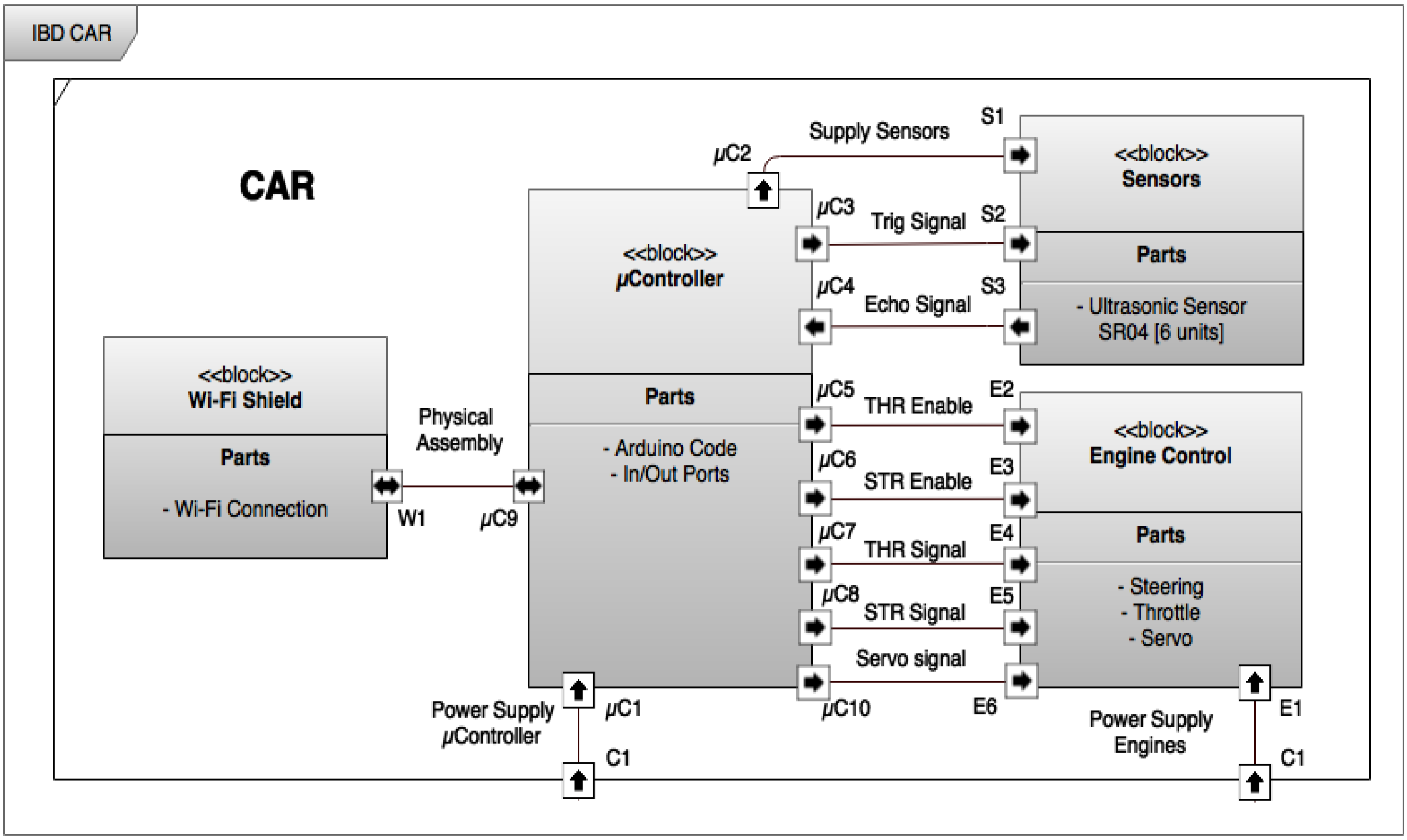
Block name	Function Description	Port Name	Direction	Comment
Car	This block consists of all HW devices of our project.	C1	Out	The car has a battery of 9,6V that we use to supply all the external devices and also the engines.
Wi-Fi Shield	Provides the Wi-Fi connection to our Arduino board.	W1	In/Out	All the Wi-Fi shield is connected to the microcontroller via direct connections
µController	Together with Phone (1) it is one of the most important devices. Controls all external components.	µC1	In	Connection where we supply our Arduino board. Directly connected with the battery.
		µC2	Out	Output port to enable the sensors
		µC3	Out	Port to activate the beginning of the sensors measurement
		µC4	In	Port where we receive data from the sensors
		µC5	Out	Port to enable the throttle control
		µC6	Out	Port to enable the steering control
		µC7	Out	We have two signals to the throttle of the car, forward and backward
		µC8	Out	We have two signals to steering of the car, right and left
		µC9	Out	Direct connection to Wi-Fi shield using the ports
		µC10	Out	Signal to move the servomotor

Sensors	We have 4 ultrasonic sensors. The model that we use is the SR04.	S1	In	Port to supply the circuits of each sensor
		S2	In	Port to activate the beginning of the sensors measurement
		S3	Out	Port where we send data to the Arduino board
Engine Control	We have two small circuits to control the throttle and steering of the car, This control depends on a PWM signal that we send from the microcontroller.	E1	In	Port where we supply the two little control circuits
		E2	In	Port to enable the throttle control
		E3	In	Port to enable the steering control circuit
		E4	In	Ports where we receive the throttle signal from the Arduino board
		E5	In	Ports where we receive the steering signal from the Arduino board
		E6	In	Port where we move the servomotor

1.6 IBD CAR

This diagram explains how our devices send and receive information between the others and shows all the signals between all the components of our HW parts.

We had to keep in mind that our Car block contains several structures where we mount all the devices and some cables to connect them together.



AARHUS UNIVERSITY

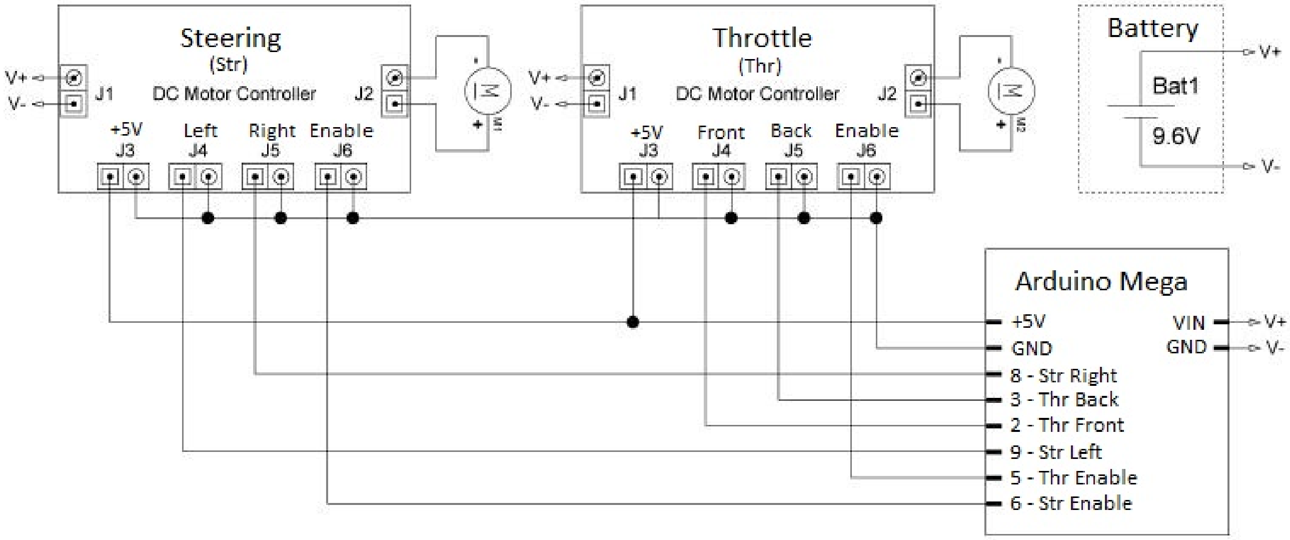
1.6.1 Explicative table IBD CAR

Name	***Description***	***Port 1***	***Port 2***	***Signal Type***
Power Supply	This connection is directly from the battery to supply with 9,6V different devices like the Arduino board and the two engines.	C1	µC1 & E1	Analog
Physical Assembly "Wi-Fi shield"	The Wi-Fi shield is mounted above the Arduino 2560 and pins fit perfectly. With this connection the Wi-Fi shield is fed and communication can begin between the two devices.	W1	µC9	Analog & Digital
Supply Sensors	We connect the Vcc connection of the sensors to a pin, in the Arduino board, that can provide 5V.	µC2	S1	Analog
Trig Signal	Signal indicating the beginning of the measurement of sensors.	µC3	S2	Digital
Echo Signal	This signal measures the time taken for the ultrasonic sound emitted by the sensor to bounce and come back. This time is operated to obtain distance.	µC4	S3	Digital
THR Enable	This connection enables the throttle control circuit from Arduino board.	µC5	E2	Digital
STR Enable	This connection enables the steering control circuit from Arduino board.	µC6	E3	Digital
THR Signal	Through this connection we send a PWM signal, from the Arduino, to control the power of the throttle in our car. We have two ports depending the direction that we want.	µC7	E4	PWM
STR Signal	Through this connection we send a PWM signal, from the Arduino, to control the power of the steering in our car. We have two ports depending which side we want to turn.	µC8	E5	PWM
Servo signal	This connection moves our servomotor using a library implemented in Arduino IDE.	µC10	E6	PWM

2 Hardware Design

2.1 Engines' control

In this section we explain how the engines of the car are connected and which signals are used to control them. First we show an overview of the connections between the different parts that take part into controlling the motors and then we show a detailed circuit diagram of the printed circuit boards (PCB) used. The following diagram was taken and modified from Autonomous Drone Control project (code 14144) Hardware architecture document.



In this picture we can see how each engine is connected to one printed circuit board (Steering or Throttle).

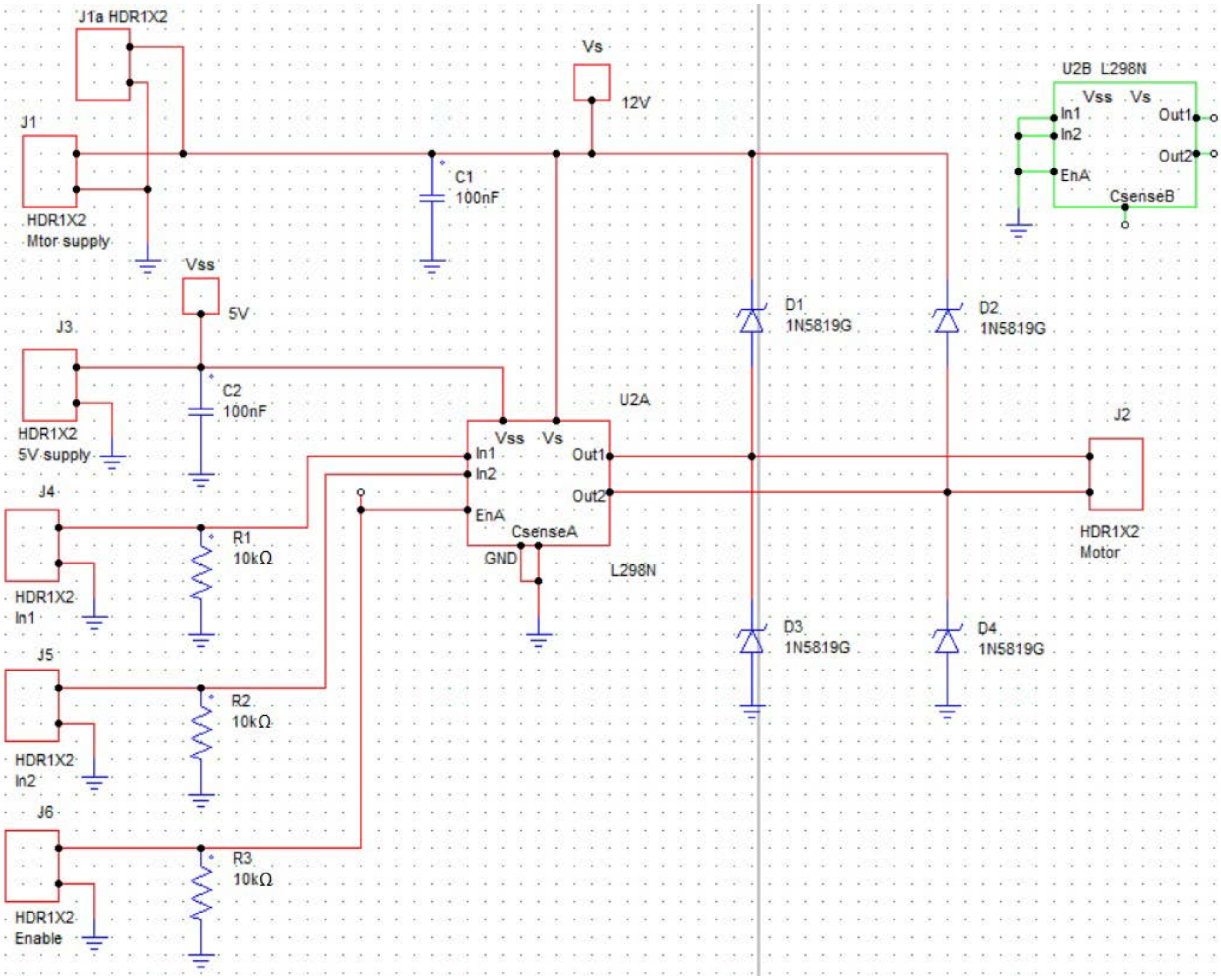
Each PCB has four input signals from the microcontroller:

- 5V
- Enable
- Rotate in one direction
- Rotate in the other direction

It can be clearly seen which signal goes out from which port from the microcontroller.

Finally, we can see that an external battery supplies the engines through the PCBs. It also supplies the microcontroller.

2.1.1 Printed circuit boards schematics



The previous figure shows a detailed circuit diagram of the used PCBs.

The PCBs used are of the L298N kind, which are motor drivers.

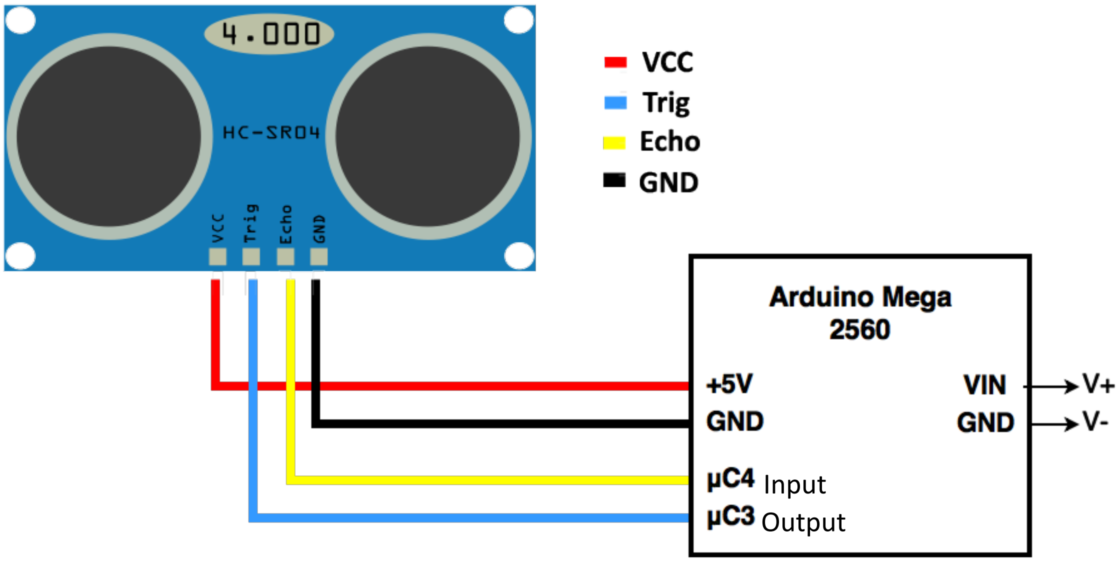
As we can see with only one of the PCBs we could have controlled both engines but only the control for one engine per PCB is used. We will not go into detail about it, because we borrowed these PCBs already made, maybe the previous users decided to make them this way because of space problem.

This diagram was inspired and created in PSIM from Autonomous Drone Control project (code 14144) Hardware architecture document.

2.2 Ultrasonic sensors

As we said previously, we are going to use the ultrasonic sensor HC-SR04

This is a small diagram showing how to connect the sensor with the Arduino board.



2.2.1 Performance

The Timing diagram is shown in the next page. You only need to supply a short 10us pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. You can calculate the distance through the time interval between sending trigger signal and receiving echo signal.

$$Distance\ (cm) = \frac{high\ level\ time\ (\mu s) \cdot velocity\ \left(343\,\frac{m}{s}\right)}{2} \cdot \frac{1\ s}{1000000\ \mu s} \cdot \frac{100\ cm}{1\ m}$$

$$Distance\ (cm) = \frac{high\ level\ time\ (\mu s)}{58{,}31\ (\frac{\mu s}{cm})}$$

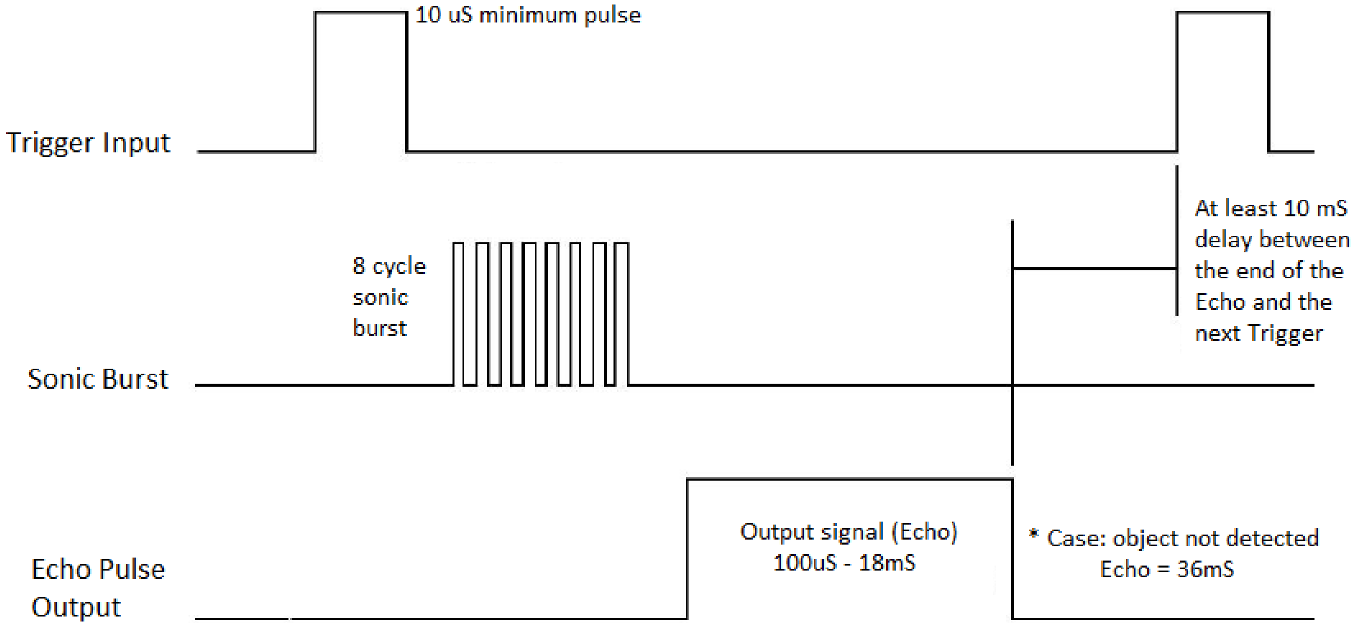
To obtain the value in centimetre we have to divide the obtained value (in microseconds) by 58,31. This value is obtained as seen above, using the standard sound velocity in the air (343 m/s) and conversion factors to obtain the desired units.

In case that we do not receive the echo, because an object is not found, the echo pulse will have a length of 36 ms.

We must leave a delay of 10 ms since the first reading until the next for circuit stability.

2.2.2 Timing diagram

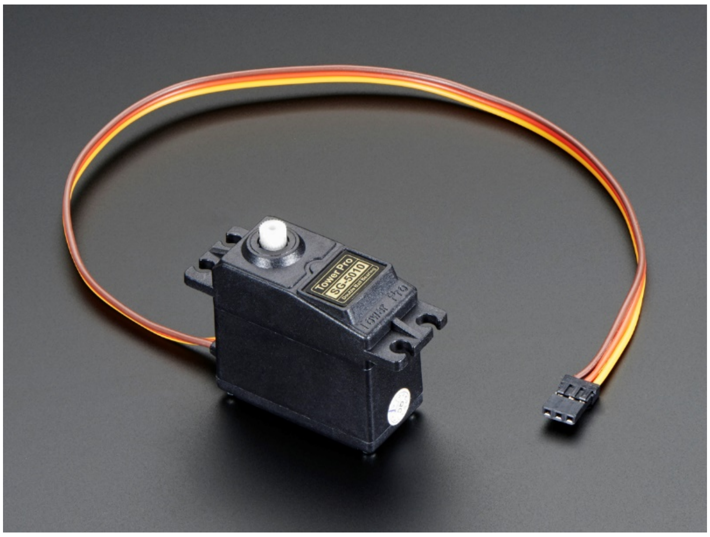
The next figure shows the signal behaviour in the Ultrasonic sensor.



2.3 Servomotor

Finally we used one servomotor to spin our sensors in the static mode of measure.

This servomotor is moved with one library implemented in the Arduino IDE, we only have to put the degree where we want to measure and wait for his movement.



The connection between the servomotor and the Arduino board is done solely with a wire, to send the PWM signal via the library “servo” in Arduino.

The circuit is supplied via 5V Port in Arduino.

Connections:

Signal (Yellow)	→	Port 13 Arduino
Vcc (Red)	→	Port 5V Arduino
GND (Brown)	→	Port 5V Arduino