Universidade do Minho
Escola de Engenharia

Remote Control

Master in Industrial Eletronics and Computers Engeneering

Embedded Systems

Authors:
Diogo Fernandes PG47150
José Tomás Abreu PG47386

Supervisors:
Prof. Dr. Tiago Gomes
Prof. Dr. Sandro Pinto

October 13, 2021

Contents

Chapter 1

Introduction

1.1   Problem Statement

Due to COVID-19 [1], many countries established mandatory lockdown, which increased the usage of entertainment platforms. Television continues to have an important role in this matter, and that's why a better user experience, through a good interface, is increasingly needed. Despite knowing that smart TV's market is growing, because of its features, a large percentage of the televisions in use are non-smart TV's [ref], which use infrared sensors to allow the user to control the TV, via a remote control. In this report, it will be analyzed and designed a remote control with the following characteristics: it must control the TV using an infrared sensor, it must be light and battery powered. It should have three buttons: one to switch the TV on/off, called "Power"; the other two buttons are used to scroll up/down and select the available channels, and they are labeled with the arrows up/down, respectively. The main goal of this project is to consolidate the methods of the Waterfall model, as it is a classic model in software development methodology, suitable for small-scale projects.

1.2   Problem Statement Analysis

In order to have a better and deeper understanding of the problem, it's essential to identify the entities involved and their relationships. Using that analysis, a system diagram can be built, figure 1.1, relating the known entities and presenting some attributes.
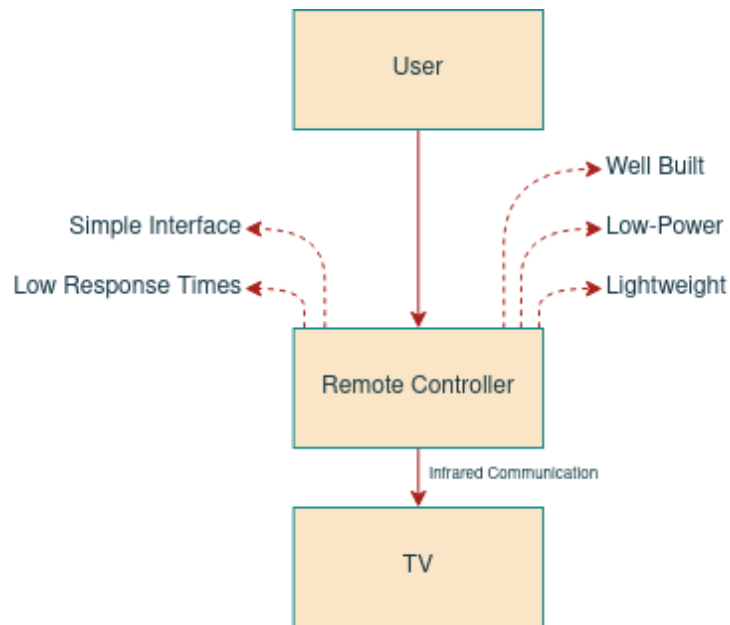
Figure 1.1: Problem Statement Analysis Diagram.

This image, figure 1.1, shows that the remote controller is the only interface between the user and the TV. The user interacts with the remote by pressing one of the existing buttons, and in turn, the remote interacts with the television via an infrared beam. Knowing that the remote controller must have a long lifespan, it must be well built, in order to resist eventual falls, and also, it must have a low consumption, in order to avoid the frequent change of batteries. For a better user experience, the remote should be lightweight, have a simple interface and respond quickly to user commands.

Chapter 2

Analysis

2.1 Market Research

2.1.1 Market Definition

A remote control, by definition, is an electronic device used to operate another device from a distance, usually wirelessly. These allow you to control different types of devices, such as televisions, air conditioning, DVD players, sound systems, among others. Remote controllers have different technologies to communicate with the devices they control, such as Bluetooth, Wi-Fi, Radio Frequency (RF) and Infrared (IR), the latter being the most used technology in remote controls.

2.1.2 Bench-Marking

Most of the time, when you buy a device that requires a remote control, it's already included. In the table below, table 2.1, there are shown some of the most popular remotes in the market right now.

Product Name	Price (€)	Weight (g)	Comm. Technology	Remote Type
Roku Voice	17,30	90,72	Wi-Fi	Dedicated
Apple TV	25,00	18,14	Wi-Fi, Bluetooth	Dedicated
DIRECTV RC73	6,05	90,71	IR, RF	Programmable
IR-1316	12,97	49,89	IR	Universal

Table 2.1: Most popular Amazon's remote controllers.

2.2 System

2.2.1 System Overview

As shown in figure 2.1, the push buttons on the remote behave as sensors, which produce an electrical signal whenever they are pressed. This electrical signal is processed by the controller, transmitting to the infrared emitter (actuator) a signal identifying the operation performed by the user, which must be received by the television in order to execute the desired operation.
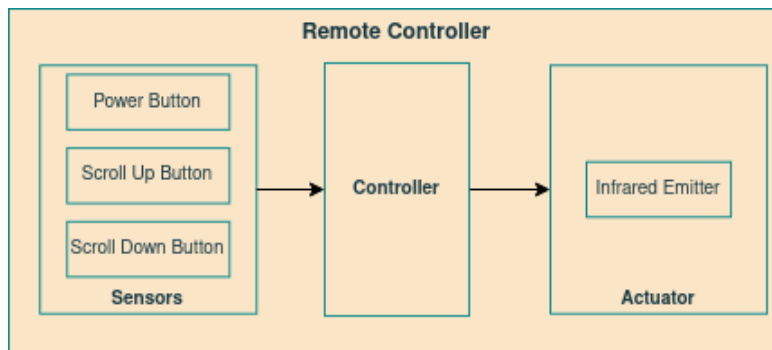

Figure 2.1: System Overview Diagram.

2.2.2 System Requirements and Constraints

Functional Requirements:

- Three push buttons: Power button, scroll up button, scroll down button.

Non-Functional Requirements:

- Lightweight;

- Low-Power consumption;

- Well built;

- Simple interface;

- Low response times.

System Constraints:

- Time constraint;

- 2 Members team;

- Infrared communication.

2.3   Hardware Architecture

The core of this system is the integrated circuit, which reads the data from the sensors (push buttons) and produces an electrical signal that identifies the operation carried out by the user. This signal is transmitted to the infrared emitter, which converts it into a light signal (infrared). The system's batteries allow powering all components. In figure 2.2 it is possible to identify all the hardware components that make up the system.
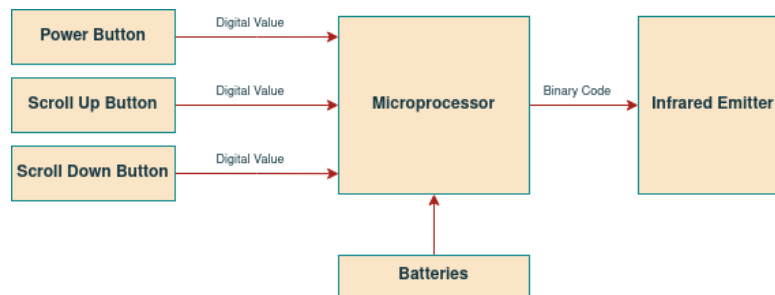

Figure 2.2: Hardware Architecture Scheme.

2.4   Use Case

A use case diagram is a graphical representation of all user's possible interactions. As shown in figure 2.3, the user is the only system entity, who can press any of the remote controller buttons. This action triggers the reading of the buttons, producing the respective code related to the pressed button, which is transmitted to the infrared emitter. The television must receive this code, performing the action required by the user.
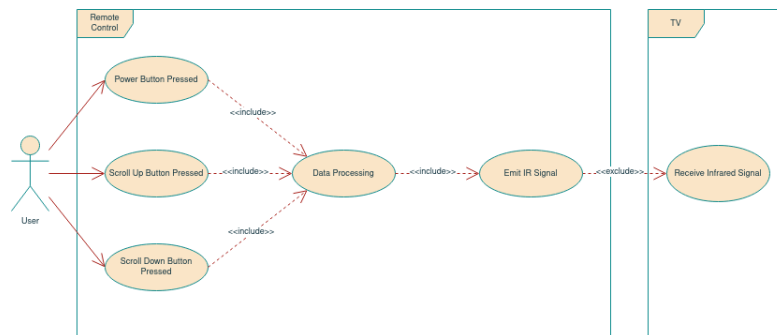
Figure 2.3: Use Case Diagram.

2.5 Sequence Diagram

A sequence diagram is another graphical representation that describes how and in what order a group of objects interacts. Figure 2.4 shows that a sequence is started when the user presses a button on the remote control. At that moment, the integrated circuit determines which button was pressed, transmitting the respective button code to the infrared emitter. In turn, the infrared emitter emits the button code via an infrared signal.
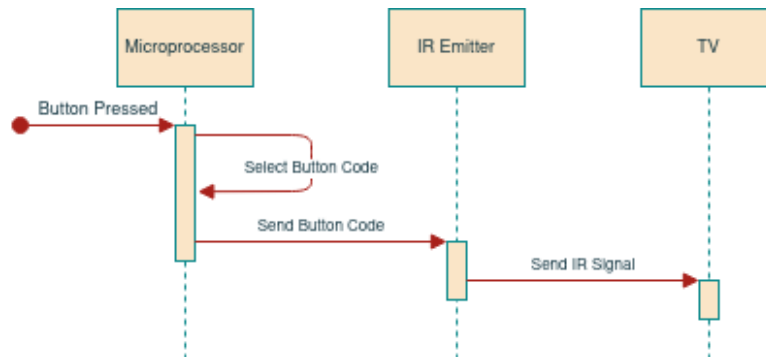

Figure 2.4: Sequence Diagram.

Chapter 3

Design

3.1   Hardware Specification

In this section it will be specified the chosen hardware and presented the interface between each of the components.

3.1.1   Microprocessor

The integrated circuit that will control the operating status of the remote will be the ATtiny85 [ref] (figure 3.1). This is a well known microcontroller due to its small dimensions, associated capabilities and its low price. Its a good fit for this system because of its low operating voltage and of its ability to read the data given by the sensors, the remote buttons, and produce an adequate response.
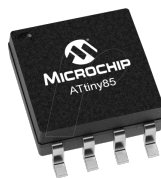

Figure 3.1: ATtiny85 10-SU [ref].

- Operating Supply Voltage: 1.8V to 5.5V;

- 8 bit Microcontroller;

- 8 pins, 6 GPIO pins;

- 8 kB of program memory size;

- 512 B data RAM size.

For programming the ATtiny85 it can be used an Arduino board, like Arduino UNO [ref], which is a one-time process.

3.1.2   IR Emitter

The system's actuator is an infrared emitter, which has been used in remote controls since the early 1980s. IR sensors are used to measure and detect infrared radiation in its surrounding environment. An IR emitter can be easily found, having the appearence represented in the figure 3.2.

Figure 3.2: IR Emitter [ref].

In this system, the IR emitter will be connected to ATtiny85 through an Output Pin.

3.1.3   Resistors

It is known that a light emitting diode (LED), like the IR emitter presented above, admits a maximum forward current. To do that, and as well as to control how much current flows through the remote button's circuit, it may be used some resistors.

3.1.4 Batteries

In order to supply the system components, focusing on the microprocessor operating supply voltage, the system may be supplied by two or three batteries AAA, having in mind that each of this batteries works within the range 0.9 V to 1.5 V.

3.2 Software Specification

To program the system's microprocessor it can be used the Arduino Software (IDE) [ref], figure 3.3, which is an open-source software, that makes it easy to write code and upload it to the board. This software can be used with any Arduino board, an so, adequate to program Arduino UNO.

Figure 3.3: Arduino Software IDE.

3.3 Flowcharts

Figure 3.4 presents an outline of the flowchart which represents the main structure of the software to be created. In each conditional structure, the status of each button is checked, that is, if any button is pressed. If no button is being pressed, nothing happens, while if a button is pressed, the code corresponding to the pressed button is sent to the IR emitter. For example, if the 'Scroll Up' button is pressed, the microprocessor will send its corresponding code via the 'send_scrollup_code' function.
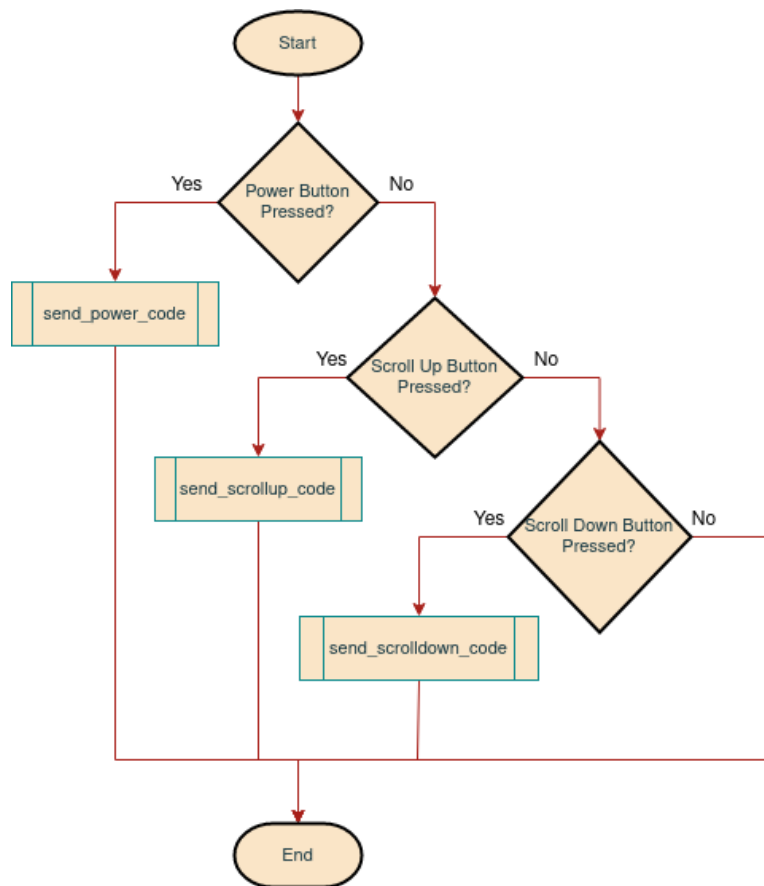
Figure 3.4: Main system flowchart.

Bibliography

[1] "Covid-19," 2021.