

Technical Documentation

IR Remote Control System for PC

Version: 1.0

Author: Sidney Himuro

License: MIT

1. Introduction

This project consists of a **microcontroller-based system using Arduino** for remotely powering computers via **infrared (IR)**.

The main technical advantages of the system are:

- **Full galvanic isolation** between the microcontroller and the motherboard, ensuring electrical safety
- **Standby power operation using ATX 5VSB**, allowing the system to work even when the PC is turned off
- **Confirmation logic for shutdown**, preventing accidental power-off events

The system safely simulates the motherboard **Power Switch (PWR_SW)** button.

2. Hardware Specifications

2.1 Main Components

- **Microcontroller:** Arduino Nano
- **Optocoupler:** PC817
 - Provides electrical isolation between Arduino and motherboard
- **IR Receiver:** VS1838B
 - Operating frequency: 38 kHz
- **Power Supply:**
 - ATX power supply 5VSB (purple wire), enabling continuous operation

2.2 Pin Configuration (Pinout)

Component / Signal	Arduino Pin	Function
VS1838B (OUT)	D2	IR signal input
PC817 (Anode)	D3	Power Switch activation
PC Status (+5V ATX)	D4	ON/OFF state monitoring
VCC	5V	Circuit power
GND	GND	Common ground
ATX Purple Wire (5VSB)	5V	Standby power
ATX Black Wire (GND)	GND	Ground reference

3. Software Configuration Procedure

The firmware is divided into **two stages**, ensuring compatibility with virtually any IR remote control.

3.1 Step 1 – Remote Control Identification (IRTest)

1. Connect the Arduino to the computer via USB.
2. Upload the code located at:

src/IRTest/

3. Open the **Serial Monitor** at **9600 baud**.
4. Point the remote control at the IR receiver and press the desired button.
5. Record the displayed values:
 - a. **Protocol**
 - b. **Address**
 - c. **Command**

These values will be used in the main firmware.

3.2 Step 2 – Main Firmware Configuration

1. Open the file:

src/IRcontrolePC/IRcontrolePC.ino

2. Insert the recorded **Protocol, Address, and Command** values into the configuration variables.
3. Compile and upload the firmware to the Arduino Nano.

4. Operation Manual

The system continuously monitors the PC state through **pin D4**, enabling safe and intelligent control logic.

4.1 Powering ON the Computer

- **Condition:** PC detected as **OFF**
- **Action:** Press the configured remote button **once**
- **Response:**
 - The Arduino activates the optocoupler for approximately **500 ms**, simulating a Power button press

4.2 Powering OFF the Computer

(*Accidental Activation Protection*)

- **Condition:** PC detected as **ON**
- **Action:** Press the configured button **three times within 5 seconds**
- **Justification:**
 - Prevents shutdowns caused by accidental presses or IR interference

5. Electrical Installation Diagram

⚠️ IMPORTANT

- All motherboard connections must be made **exclusively** to the **Power Switch (PWR_SW)** pins.

5.1 Electrical Isolation

- The **PC817 output terminals (collector and emitter)** must be connected to the two PWR_SW pins.
- Polarity is generally **not critical**, as the original button works as a dry contact.

5.2 PC State Signal

- The **+5V (red wire)** from the ATX power supply can be used to detect when the PC is powered ON.
- Recommended:
 - Use of a **voltage divider**, or
 - Prior validation of signal stability

5.3 Continuous Power Supply

- The Arduino **must be powered by the ATX 5VSB (purple wire)**.
- Otherwise, the system will shut down together with the PC and lose the ability to power it on remotely.

6. Troubleshooting

PC does not power ON

- Verify voltage presence on the **5VSB (purple wire)** while the PC is off.
- Check continuity of the **PC817 optocoupler circuit**.

Remote control does not respond

- Ensure the **VS1838B IR receiver** is not obstructed.
- Avoid direct exposure to strong sunlight.
- Repeat the identification procedure using **IRTest**.

PC shuts down unexpectedly

- Check for electrical noise on **pin D4**.
- A **10 kΩ pull-down resistor** on the state monitoring pin is recommended.