# Circuit Documentation

## Summary

This circuit is designed to interface an Arduino Mega 2560 with various components including a DS3231 Real-Time Clock (RTC), an IR Receiver, a 5V 8-Channel Relay module, pushbuttons, linear actuators, and power sources. The Arduino Mega 2560 serves as the central processing unit, controlling the relays which in turn drive the linear actuators. The DS3231 RTC provides timekeeping functionality, while the IR Receiver allows for remote control capabilities. Pushbuttons are used for manual input, and the circuit is powered by 12V batteries.

## Component List

### Arduino Mega 2560

* Microcontroller board based on the ATmega2560
* It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

### DS3231 RTC

* A precise real-time clock module with an integrated temperature-compensated crystal oscillator (TCXO) and crystal.
* It communicates with the Arduino via I2C.

### IR Receiver

* An infrared receiver that can decode IR signals from a remote control.
* It is connected to one of the digital pins of the Arduino.

### 5V 8-Channel Relay

* A relay module with 8 channels that can be controlled independently.
* Each relay can be used to switch higher voltage loads.

### Pushbuttons

* Simple push-to-make buttons for user input.
* They are connected to the digital pins of the Arduino.

### Linear Actuators

* Devices that convert electrical energy into linear motion.
* They are controlled by the relays and are powered by the 12V batteries.

### 12V Batteries

* Provide the power source for the linear actuators and the relay module.

### 2.1mm DC Barrel Jack

* A power connector that allows the 12V battery to connect to the circuit.

## Wiring Details

### Arduino Mega 2560

* GND connected to the ground of DS3231 RTC, IR Receiver, and both pushbuttons.
* 5V supplies power to the 5V 8-Channel Relay, DS3231 RTC, IR Receiver, and both pushbuttons.
* D2 PWM to IN1 on the 5V 8-Channel Relay.
* D3 PWM to IN2 on the 5V 8-Channel Relay.
* D7 PWM to IN3 on the 5V 8-Channel Relay.
* D8 PWM to IN4 on the 5V 8-Channel Relay.
* D4 PWM to IN5 on the 5V 8-Channel Relay.
* D5 PWM to IN6 on the 5V 8-Channel Relay.
* D10 PWM to IN7 on the 5V 8-Channel Relay.
* D11 PWM to IN8 on the 5V 8-Channel Relay.
* D21/SCL to SCL on DS3231 RTC.
* D20/SDA to SDA on DS3231 RTC.
* D12 PWM to DATA on IR Receiver.
* D9 PWM to Pin 3 on the first pushbutton.
* D6 PWM to Pin 3 on the second pushbutton.

### DS3231 RTC

* GND to ground.
* VCC to 5V from Arduino.
* SCL to D21/SCL on Arduino.
* SDA to D20/SDA on Arduino.

### IR Receiver

* DATA to D12 PWM on Arduino.
* VCC to 5V from Arduino.
* GND to ground.

### 5V 8-Channel Relay

* VCC to 5V from Arduino.
* GND to ground.
* IN1 to D2 PWM on Arduino.
* IN2 to D3 PWM on Arduino.
* IN3 to D7 PWM on Arduino.
* IN4 to D8 PWM on Arduino.
* IN5 to D4 PWM on Arduino.
* IN6 to D5 PWM on Arduino.
* IN7 to D10 PWM on Arduino.
* IN8 to D11 PWM on Arduino.
* NC (Normally Closed) pins are not connected.
* NO (Normally Open) pins are connected to the positive side of the 12V battery.
* C (Common) pins are connected to the positive or negative side of the linear actuators, depending on the desired direction of motion.

### Pushbuttons

* One side connected to 5V from Arduino.
* The other side connected to D9 PWM and D6 PWM on Arduino respectively.
* Ground connected to the ground.

### Linear Actuators

* Positive and negative pins connected to the C and NO pins of the 5V 8-Channel Relay respectively.

### 12V Batteries

* One battery's positive terminal connected to the center pin of the 2.1mm DC Barrel Jack and NO on the 5V 8-Channel Relay.
* The negative terminal connected to the sleeve pin of the 2.1mm DC Barrel Jack and NC on the 5V 8-Channel Relay.

### 2.1mm DC Barrel Jack

* center connected to the positive terminal of the 12V battery.
* sleeve connected to the negative terminal of the 12V battery.

## Documented Code

void setup() {  
 // put your setup code here, to run once:  
  
}  
  
void loop() {  
 // put your main code here, to run repeatedly:  
  
}

The provided code is a template for the Arduino sketch and does not contain any functional code. It consists of two primary functions: setup(), which runs once when the microcontroller is reset or powered on, and loop(), which runs continuously after setup() completes. The user is expected to fill in these functions with code that initializes the hardware components and defines the main behavior of the circuit.