SmartSwitch

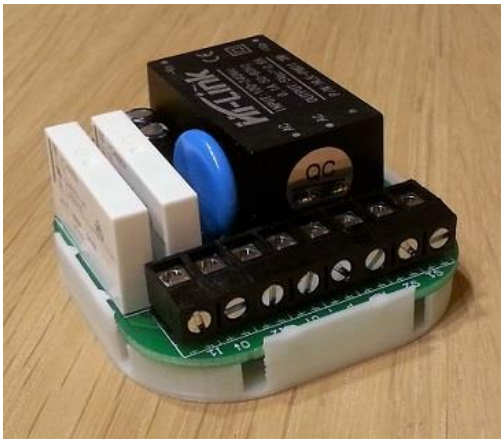
Flush-mounted Module to control shutters / realize an intermediate switch with ESP8266

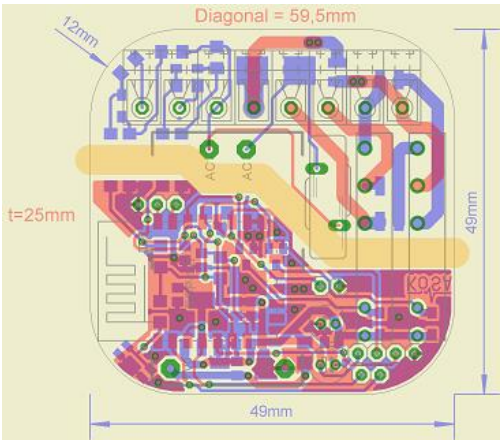
Version History

- SmartSwitch V1.2 04/2018 1st official public Version
- SmartSwitch V1.3 05/2018 ADD: Additional Fuse (F2, 4A) for consumer current
- **Software now avaliable for ArduinoIDE (including OTA)**

Vision

During the construction of our new house, we've decided to install motorized roller shutters. To control them, we first used only standard shutter-push-buttons, right beside the correspondent window. As second step a *smart solution* should be established.



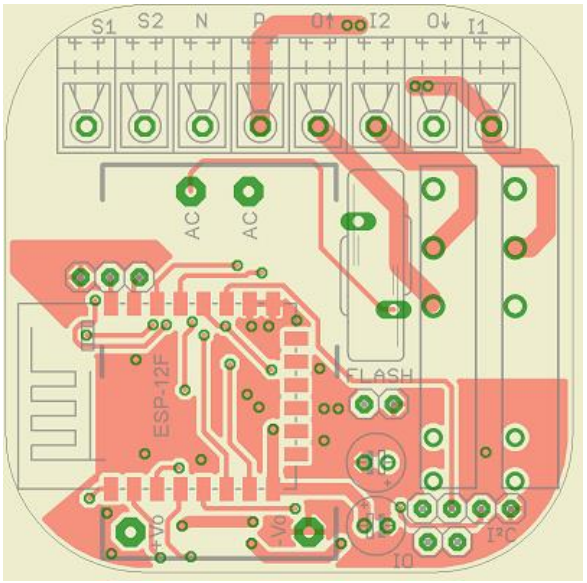


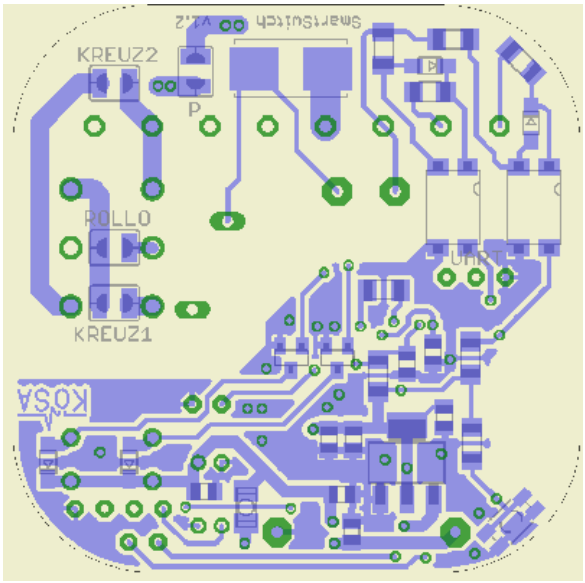
Inspired by Leo-Andres Hofmann's "230V I/O Modul für ESP8266" (LUANI), I've decided to develop my own module to sense the shutter-push-buttons and control the motorization. By its dimensions, the module perfectly fits into a standard round In-Wall-Pattress (inner diameter: 60mm). If its a so called "deep" pattress, the shutter-push-buttons can also be installed on top of the module and there is enough space left for wires and terminals.

Basis of the module is a ESP-12F, an integrated WLAN-Module which controls two changeover relays to move the shutters.

The operation of the module can either be controlled by the already existing push-buttons, or via MQTT-Messages by WLAN, central controlled via OpenHab2, running on my Raspberry Pi Zero W. Therefore, a movement of the shutters regarding time or special events, for example time of dusk and down (Astro Bindings) can be realized.

Also an automatized shadowing regarding sun position, global radiation and inside / outside temperature will be possible.





Hardware

Beside the WLAN-Module and the relays (contact rating 1500VA, 6A/250V), a power supply consisting of a AC/DC-Converter HI-Link HLK-PM01 (5V/600mA) to serve the relays and a voltage regulator TS1117 (LowDrop 3V) to serve the remaining circuit, a 2-chanel power sense circuit to detect 230V levels for the controller and interfaces like UART, I^2C and analog/digital IO are included on the board. A push button for hardware reset is also implemented on bottom side of the PCB.

Three soldering jumpers give the opportunity to decide if the relays should act as an intermediate switch to implement the module in an already existing light-installation or to act as a roller-shutter-controlling device. Therefore, the first relay powers the motor during the second relay controls moving direction. Via a fourth soldering Jumper it is possible to supply mains voltage to the relays, so for roller shutter function, there is no need to connect phase two times, for supplying the module and supplying the motor.

The power supply of the module is secured by a SMD-Fuse (500mA) and a varistor (275V) against overload.

Software

In addition to the firmware (NodeMCU) there are mainly four LUA-Scripts on the Module.

The Init-File tries to connect to the access point whose SSID and password is stored at the ESP. If the connection couldn't be established, the ESP himself opens an access point and starts an end user setup to give the user the possibility to enter SSID and password of an already existing access point (Router).

If the connection is established, two more software modules will be started. The MQTT-Module starts to configure the implemented MQTT-Client, opens a connection to the MQTT-Broker which is defined (IP, Port) in the MQTT.ini file and provides several procedures for MQTT communication. The INI-Handling-Module provides two more procedures for easy INI-File access by using "Key" and "Value" data.

Finally, the main program will be started.

☝ *UPDATE*

Now I'd also written a new software for ArduinoIDE. The main advantage of this new sketch is the possibility of updating the software via OTA (Over the Air update via WiFi). After flashing the sketch via serial adapter for the first time, the module can be found and flashed via ArduinoIDE OTA-Functionality.

Additionally there are a few smaller upgrades and BugFixes implemented in the new software.

The functionality of the Arduino-Software is mostly the same as the LUA-Skript. The module tries to connect to the last stored access point. If no connection could be established the module starts its implemented enduser setup. SSID and password of the AP and also the site of usage and MQTT-borker IP can be configured. After this initialisation the module starts its normal operation and can be either controlled via the already existing push buttons or MQTT-Commands.

Function

During normal operation, the main program offers the opportunity to control the module via the shutter-push-buttons by using both of the power sense inputs. (Only LUA-Skript: If both buttons were pushed synchronously the module changes to teaching mode.) Via MQTT-Command "TEACH", the module changes to teaching mode. Because of the fact that simple shutter motors neither feedback their position nor the state of their limit switches, the position of the shutter can only be traced by the time, the motor is in motion. Therefore, in teaching mode, the time to reach both final positions can be stored in module by closing and reopening the shutter completely. After that, a short push of a button results in moving to respective end position. If the button will be pressed longer then 1s the shutter moves as long as the

button is pressed or final position is reached. By releasing the button, the current position of the shutter will be calculated regarding the time, the motor was in motion and will be stored in the module. Thus offers the opportunity to start motion from any position of the shutter or moving the shutter to a particular position (e.g. closing until only shutter slots are open).

But not only manual operation by pushing a button is possible. As mentioned, the module acts also as a MQTT-Client. Via messages to the MQTT-Topic (LUA-Skript: which is also settled in MQTT.ini file; ARDUINO-Skript: /SmartSwitch/SmartSwitch_<SITE or MAC>/), motion to the final position is possible, either by command “UP” or “DOWN”. By sending a value from 0 to 100 the shutter moves to the correspondent position in percent. If the desired position is reached or the shutter reaches end position, the current position will be also send as MQTT-Message to the broker (ARDUINO: Also the RSSI of the module will be sent). Therefore, a visualization of the current position e.g. with HabPanel von OpenHab is possible.

Safety

By operating the module at mains voltage, the active parts and pins should be covered against directly and indirectly touching. Therefore, I’ve designed a Bottom-Coverage for the module with FreeCAD and printed it with my Anet A6 3D-Printer. As positive side effect of the coverage, all SMT-Components on bottom side are protected against mechanical damages.

Safety Disclaimer

Building and usage at your own risk. SmartSwitch uses potentially lethal mains voltages.

If you are a beginner in electronics or don't have the proper experience to work with mains voltage, consult a professional! I am not liable for any mishaps that may occur due to misuse of SmartSwitch or electrical system (including mains power) including but not limited to: errors or omissions, loss of life, or property damage. Take all safety precautions as necessary. You are fully responsible for your actions, perform them at your own risk!