European Research Studies Journal
Volume XXVII, Issue S3, 2024

pp. 1020-1026

A Cost-Effective Solution for Home Automation

Submitted 10/10/24, 1st revision 25/10/24, 2nd revision 01/11/24, accepted 05/12/24

Mariusz Śniadkowski[1], Dorota Wójcicka-Migasiuk[2], Agnieszka Gandzel[3]

Abstract:

Purpose: *Both photovoltaics and home automation solutions are growing in popularity year on year. A modern smart home solution combined with photovoltaics is not only environmentally friendly, but also economical. Home automation allows maximum savings and the most efficient use of photovoltaics. It guarantees convenience and comfort in everyday use. It also fits in with the now popular 'less waste' or 'zero waste' trend, which strives for the efficient use of raw materials.*
Design/Methodology/Approach: *The concept of home automation has gained considerable traction in recent times, with a plethora of solutions offered by various companies.*
Findings: *However, the cost associated with these solutions is often a deterrent. The devices and solutions proposed by companies for home automation systems are typically based on IEEE 802.11g WiFi communication, which is dependent on the strength of the wireless signal and is characterised by high latency. Conversely, utilising communication based on the I2C protocol to control home appliances has the potential to significantly reduce the cost of building the system.*
Practical implications: *As a result of conducting the research, it is possible to identify and present some recommendation for reduces the cost of building the system home automatisation.*
Originality value: *For the purposes of this article, a multifaceted, synthetic and critical analysis of data available in the source literature was carried out.*

Keywords: *Home automation, cost-effective solutions, smart home technology, IoT, energy efficiency.*

JEL classification: *F52, D12, L68, O33, Q55, R31.*

Paper type: *Research article.*

[1] *Prof., dr hab., Lublin University of Technology, Poland, ORCID: 0000-0001-7561-4224, m.sniadkowski@pollub.pl;*
[2] *Prof., dr hab., Lublin University of Technology, Poland, ORCID: 0000-0003-0856-716X, d.wojcicka-migasiuk@pollub.pl;*
[3] *Ph.D., Lublin University of Technology, Poland, ORCID: 0000-0002-7887-8636, a.gandzel@pollub.pl;*

1. Introduction

The core of the solution presented is the free and open Domoticz system (Setz *et al.*, 2021), which runs on a wide range of software and hardware platforms such as Linux, Windows, Arm 32 and 64 bit. However, for reasons of energy efficiency and reliability, it is most commonly installed on single-board computers (SBCs) such as Raspberry Pi, Orange Pi, Banana Pi, etc.

Domoticz also has a dedicated mobile application for tablets and phones, which can be used to conveniently control various devices. Domoticz communicates via a USB port with an Arduino, which manages the Inter-Integrated Circuit (I^2C) bus and other connected sensors.

2. Literature Review

A protocol is a way of communicating between computers, network devices, or computers and terminals, and captures the rules and procedures for accessing the cabling system. Network communication protocols are designed to allow computers to communicate with each other and to transfer data efficiently and reliably. Modern home automation systems are most often based on IEEE 802.11g WiFi communication (ElShafee and Hamed, 2012), which is dependent on the strength of the wireless signal and is characterised by high operating delays.

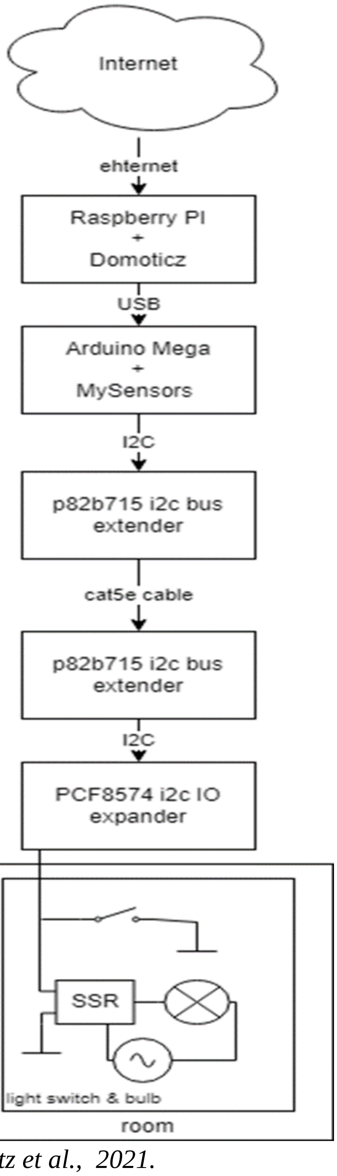
The I^2C protocol is included in almost every modern processor or minicomputer used for smart home management (Addabbo *et al.*, 2019). Using wired communication therefore significantly increases reliability, speed and reduces the cost of building a home system. The I^2C synchronous serial interface is used to communicate with various sensors, memory chips, digital switches, LCD displays, etc.

An important limitation of the I^2C bus is its maximum line capacity, which in practice limits its length to a few metres between sensor and controller (He, Chen, Li and Yi, 2019). However, the use of an extender increases the length of the line.

3. Research Methodology

Installing Domoticz (Aiello and Dustdar, 2008) on a Raspberry Pi is a relatively simple task and will only take a few minutes of your time. You will need to download the Raspberry Pi Imager from https://www.raspberrypi.com/ and burn the Lite OS version to an SD card. After booting the minicomputer from the SD card, log in with login: pi and password: raspberry. To install Domoticz, issue the command curl -sSL install.domoticz.com | sudo bash.

Figure 1. *A visual representation of the approach that has been taken in this work.*

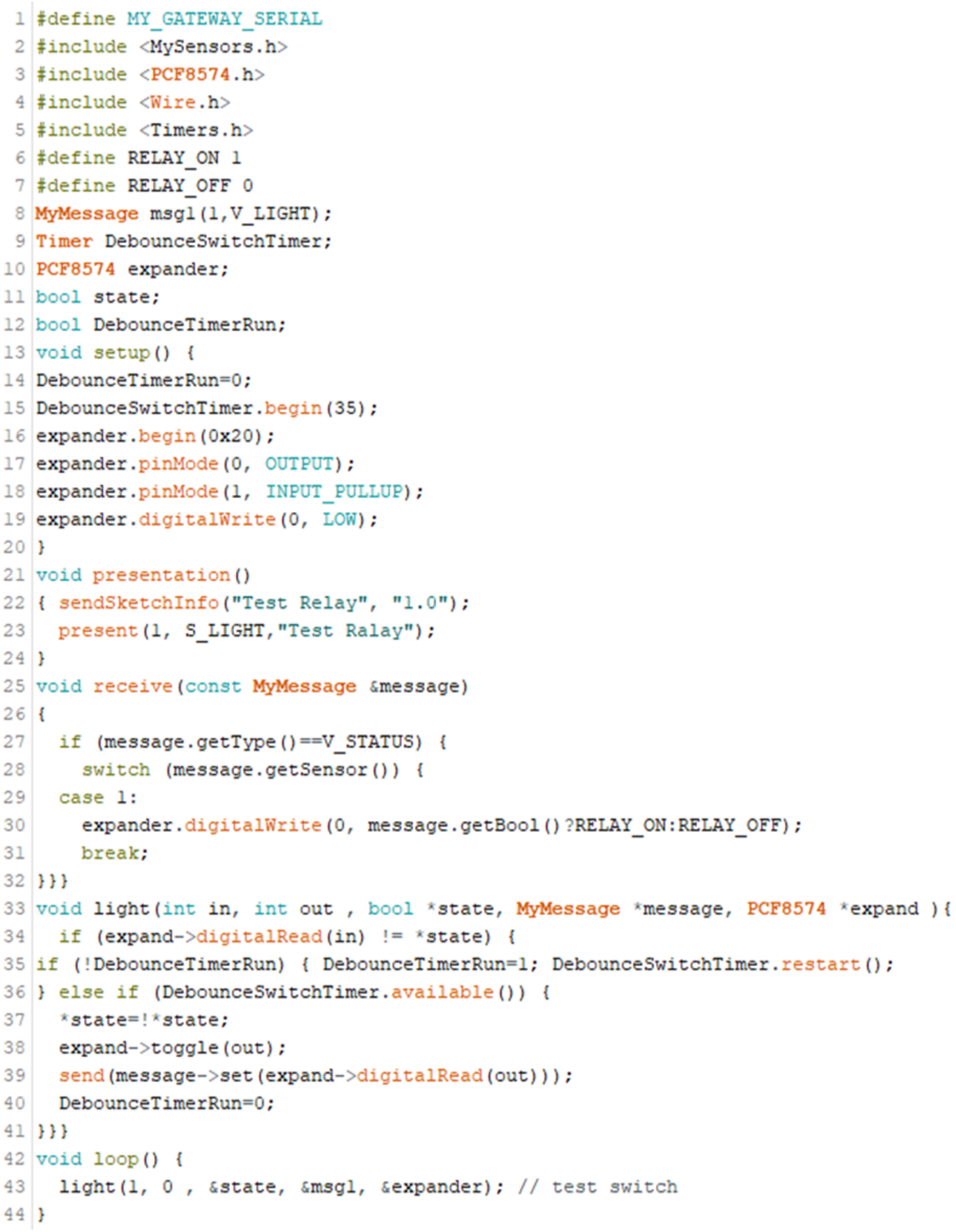


Source: *Own study based on Setz et al., 2021.*

4. Arduino - My Sensors Library

We can install the MySensors library by selecting Tools -> Manage Libraries, then we need to search for MySensors, select the latest version and press the Install button. In this project the library will be used for USB communication between Arduino and Domoticz. Using the library consists of writing two functions presentation() and receive(const MyMessage). The first function presentation informs Domoticz about the supported devices, while the second function receive provides the operation, which in this case is to switch the SSR relay on and off.

We can see the sample Arduino code in the example:



```
#define MY_GATEWAY_SERIAL
#include <MySensors.h>
#include <PCF8574.h>
#include <Wire.h>
#include <Timers.h>
#define RELAY_ON 1
#define RELAY_OFF 0
MyMessage msg1(1,V_LIGHT);
Timer DebounceSwitchTimer;
PCF8574 expander;
bool state;
bool DebounceTimerRun;
void setup() {
DebounceTimerRun=0;
DebounceSwitchTimer.begin(35);
expander.begin(0x20);
expander.pinMode(0, OUTPUT);
expander.pinMode(1, INPUT_PULLUP);
expander.digitalWrite(0, LOW);
}
void presentation()
{ sendSketchInfo("Test Relay", "1.0");
  present(1, S_LIGHT,"Test Ralay");
}
void receive(const MyMessage &message)
{
  if (message.getType()==V_STATUS) {
    switch (message.getSensor()) {
  case 1:
    expander.digitalWrite(0, message.getBool()?RELAY_ON:RELAY_OFF);
    break;
}}}
void light(int in, int out , bool *state, MyMessage *message, PCF8574 *expand ){
  if (expand->digitalRead(in) != *state) {
if (!DebounceTimerRun) { DebounceTimerRun=1; DebounceSwitchTimer.restart();
} else if (DebounceSwitchTimer.available()) {
  *state=!*state;
  expand->toggle(out);
  send(message->set(expand->digitalRead(out)));
  DebounceTimerRun=0;
}}}
void loop() {
  light(1, 0 , &state, &msg1, &expander); // test switch
}
```

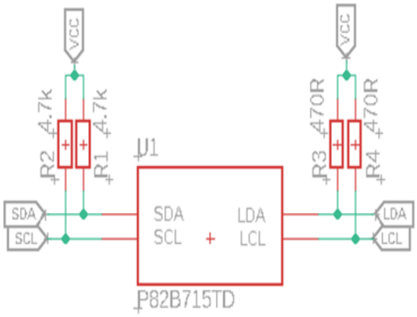
The PCF8574 and Timers libraries (available at
https://github.com/skywodd/pcf8574_arduino_library
and https://github.com/nettigo/Timers)
have been used here to simplify the code. The PCF8574 library allows easy control of individual expander pins, while the Timers library was used to time the vibration of the wall switch contacts.

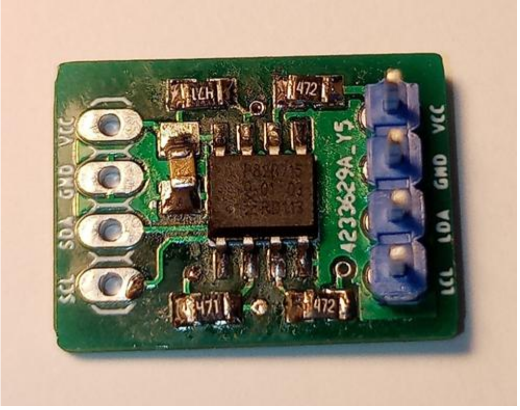
The light function checks the status of the expander port to which the wall switch is connected and, if it changes, also changes the status of the next potentiometer in the expander to which the SSR relay controlling the room lighting is connected. This function also informs Domoticz of the change so that the status of the corresponding switch in the application is updated.The Arduino programmed in this way is automatically detected by Domoticz and all connected inputs and actuators are immediately visible.

5. Practical and Economical Implementation

To increase the I^{2}C transmission range, the project uses an extender. This is a low cost chip in an SOP8 package. It costs less than a dollar each on a popular Chinese sales site. The circuit's application scheme boils down to the use of four pull-up resistors.

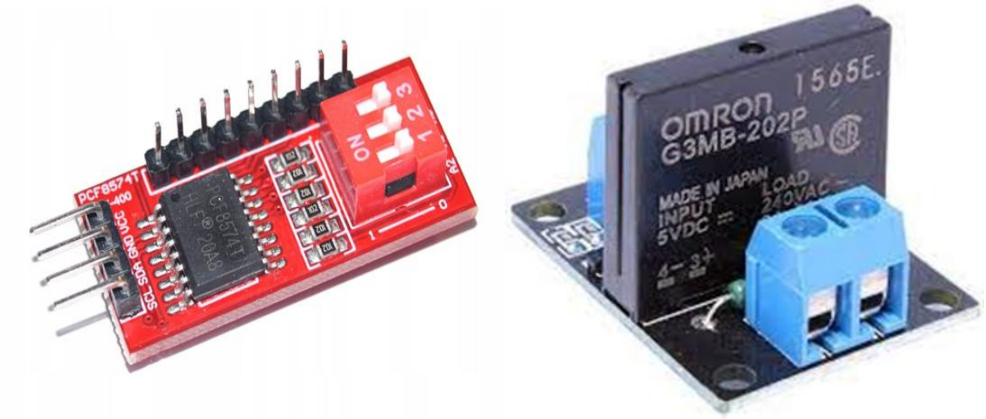
Image credit: Mariusz Śniadkowski.





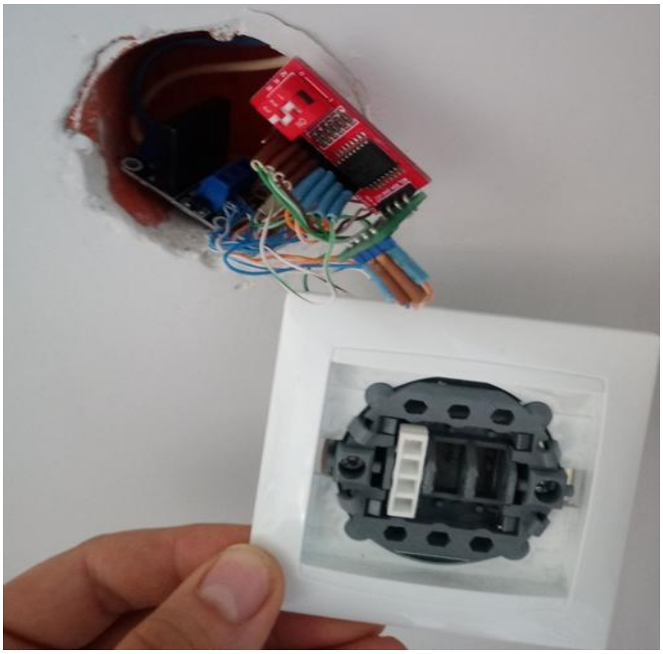
As well as the P82B715 extender, there is a ready-to-use module using the PCF8574T chip, which is also available for less than a dollar.

Image credit: Mariusz Śniadkowski.



The whole system, including the SSR relay, fits easily into a deep fi 60mm socket. Although the circuit does not have built-in pull-up resistors for the expander ports, in practice the design shown will work correctly without them.

Image credit: Mariusz Śniadkowski.



6. Conclusions

The design outlined above represents a practical, cost-effective and straightforward-to-deploy smart home system. The system is distinguished by its straightforward programming, extensive component availability, and adaptability to varying requirements. The practical implementation employs a pipe topology comprising pairs of P82B715 extenders and PCF8574T modules, with three extenders and three modules in each pair.

Such a circuit is stable and may be employed, for instance, to regulate domestic lighting or other devices. It is possible to extend the design to include additional sensors, such as those for measuring temperature or humidity. In practice, the only limit is the imagination of the system designer.

Electric cars or plug-in hybrids, which until recently were only achievable for lovers of expensive technical innovations, are becoming a solution available to everyone. In the not-too-distant future, an electric car charger will be a permanent fixture in the garage or a parking space at home.

References:

Addabbo, T., Fort, A., Mugnaini, M., Parrino, S., Pozzebon, A., Vignoli, V. 2019. Using the I2C bus to set up Long Range Wired Sensor and Actuator Networks in Smart Buildings. 4th International Conference on Computing, Communications and Security (ICCCS), pp. 1-8. doi: 10.1109/CCCS.2019.8888085.

Aiello, M., Dustdar, S. 2008. Are our homes ready for services? A domotic infrastructure based on the Web service stack. Pervasive and Mobile Computing, 4(4), 506-525.

ElShafee, A., Hamed, K. 2012. Design and implementation of a WIFI based home automation system. International Journal of Computer and Information Engineering, 6(8), 1074-1080.

He, H., Chen, Y., Li, Z., Yi, W. 2019. I2C: Joint Intra-Packet and Inter-Packet Coding for Reliable Cross-Technology Communication. In: IEEE Communications Letters, vol. 23, no. 6, pp. 1085-1088. doi: 10.1109/LCOMM.2019.2911080.

Setz, B., Graef, S., Ivanova, D., Tiessen, A., Aiello, M. 2021. A Comparison of Open-Source Home Automation Systems. In: IEEE Access, vol. 9, pp. 167332-167352. doi: 10.1109/ACCESS.2021.3136025. Keywords: {Open source software;Smart homes;Costs;Standards;Protocols;Feature extraction;Computer architecture;Home automation systems;Internet of Things;open-source projects;home automation architecture}.