

W systemie zostały zastosowane czujniki zbliżeniowe magnetyczne MC-38. Pozwalają one określić pozycję drzwi albo okien. Część z przewodami należy umieścić na framudze a część swobodną na drzwiach lub oknach. Obwód w części z przewodami jest domyślnie rozwarty, a po zbliżeniu do magnesu zostaje zamknięty. Czujnik powinien być podłączony do wyprowadzenia GPIO i do masy, aby wykrywać zamknięcie i przerwanie obwodu. Raspberry Pi umożliwia podłączenie każdego wyprowadzenia poprzez wbudowany rezystor podciągający do napięcia zasilania (Vcc). Dzięki temu można wykrywać otwarcie okien lub drzwi jako oddalenie czujników i przerwanie obwodu – zmianę stanu odczytywaną na wyprowadzeniu z wysokiego na niski. W wykonanym prototypie dłuższy przewód podłączono do wyprowadzenia GPIO, a krótszy do masy na płytce prototypowej.

Korzystanie z wyprowadzeń GPIO w programie języku Python jest możliwe na komputerze Raspberry Pi przy użyciu pakietu RPi.GPIO. Umożliwia on odczyt stanu wyprowadzeń. Konfiguracja obsługi GPIO rozpoczyna się od określenia sposobu numeracji wyprowadzeń. Dostępne są dwie możliwości: GPIO.BOARD oraz GPIO.BCM. Pierwszy odnosi się do fizycznej lokalizacji wyprowadzeń na płytce drukowanej. Drugi sposób oznacza numeracją kanałów system-on-chip (SOC) firmy Broadcom, który jest użyty w komputerze Raspberry Pi 3B. Ze względu na korzystanie z płytki prototypowej oraz modułu Proto Pi Plus, który korzysta z oznaczeń odpowiadających kanałom SOC w projekcie użyto tego sposobu numeracji.

Inicjalizacja obsługi GPIO wywoływana jest w funkcji:

```
def init_gpio():  
    GPIO.setmode(GPIO.BCM)
```

Wykonywana jest ona na początku działania programu nadzor.py w funkcji main(). Obsługa poszczególnych czujników odbywa się w ramach obiektów klasy Grupa. Użycie rezystora podciągającego jest wywoływane w programie nadzor.py w konstruktorze obiektu klasy Grupa w następujący sposób.

```
GPIO.setup(self.czujnik.gpio, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

self.czujnik.gpio - to atrybut przechowujący numer wyprowadzenia GPIO. GPIO.IN - konfiguruje wyprowadzenie jako wejście. *pull_up_down* = *GPIO.PUD_UP* - to flaga określająca zastosowanie rezystora podciągającego.

Wymogiem technicznym systemu jest również obsługa czujników przy użyciu magistrali I2C. W ramach wykonanego systemu została zapewniona obsługa czujników temperatury i wilgotności. Wybrany został czujnik Si7021 firmy Adafruit. Zapewnia on pomiar temperatury w zakresie od -10 do 85°C oraz wilgotności względnej od 0 do 80%. Są to wartości wystarczające do pracy w warunkach pracy w zamkniętym pomieszczeniu. Wymogiem systemu jest również obsługa wielu czujników i kamer. Czujniki Si7021 posiadają niezmienny adres równy 0x70. Nie mają one możliwości programistycznej lub sprzętowej zmiany adresu. Oznacza to, że po podłączeniu ich do jednej magistrali I2C nie byłoby możliwości jednoznacznego określenia, z którego z urządzeń został odczytany wynik pomiaru. Rozwiązaniem tej sytuacji jest zastosowanie multipleksera I2C. Przykładem takiego urządzenia jest TCA 9548A firmy Adafruit. Pozwala on na podłączenie do 8 urządzeń korzystających z magistrali I2C.

Konfiguracja odbywa się w następujący sposób. Do multipleksera przesyłany jest 8-bitowy kod odpowiadający numerowi kanałowi. Aktywacja kanału oznacza ustawienie bitu o pozycji równej numerowi kanałowi jako 1. Po wysłaniu komendy i znaku STOP kanał jest aktywowany. Kolejne komendy można adresować, używając adresu czujnika temperatury. Pierwszą komendą, którą należy wysłać to komenda pomiaru wilgotności

względnej. Dostępne są dwa tryby pomiaru tej wielkości: Hold Master Mode oraz No Hold Master Mode. Pierwszy z nich oznacza, że urządzenie nadrzędne wysyła żądanie pomiaru wilgotności. Urządzenie podrzędne potwierdza otrzymanie żądania i dokonuje pomiaru. Wymaga to zastosowania rozciągania zegara (clock stretching), które polega na utrzymywaniu przez urządzenie podrzędne linii zegarowej SCL w stanie niskim. Dzieje się to do momentu zakończenia pomiaru przez urządzenie i wystawieniu jego wyniku do rejestru. Drugi tryb różni się tym, że po potwierdzeniu otrzymania żądania wystawiany jest symbol zakończenia komunikacji. Znając maksymalny czas konwersji pomiaru, urządzenie nadrzędne może odczytać wynik z rejestru po jego zakończeniu.

Do obsługi żądań konieczna była odpowiedni pakiet. Pierwszym zastosowanym pakieciem był pakiet python-smbus. Korzysta on ze sterownika wbudowanego w jądro systemu Linux. Została przeprowadzona próba komunikacji w trybie Hold Master Mode ze względu na dostępne komendy protokołu I2C. Zakończyła się ona błędem o kodzie io errno5. Wynikał on z tego, że pakiet smbus obsługuje magistralę I2C zgodnie ze standardem SMBus. Posiada on bardziej ścisłe reguły dotyczące czasu trwania transakcji na magistrali. Przy użyciu pakietu nie było możliwe zastosowanie trybu No Hold Master Mode, ponieważ nie dostarczał on komendy odpowiadającej odczytowi rejestru urządzenia o wskazanym adresie. Wszystkie dostępne komendy zawierały zapis do urządzenia przed odczytem.