BMA400:

* Tryb pracy: low power – po wykryciu ruchu automatycznie przechodzi do normal.
* Zasada działania: czujnik reaguje na ruch w jakiejkolwiek osi, trzeba dostosować, żeby nie zbierał małych drgań, jak np. wiatr zawieje. Jak wykryje ruch, to wybudza STM
* Dane są przesyłane przez I2C do STM32

STM32:

* Tryb uśpienia: STANDBY – wybudzanie sygnałem z INT czujnika.
* Przeliczone procenty są przesyłane na bieżąco do ESP przez UART.
* Trzeba np. analizatorem stanów logicznych sprawdzić czy po wybudzaniu się nie powoduje jakichś skoków na PA5 (EN\_IO) i ew. skorygować w programie, bo to by bardzo dużo energii pobierało.

ESP8266:

* Zawsze wyłączony. Aktywowany przez STM32 na pinie EN\_ESP podczas transmisji.
* Przesyła procent otwarcia do drugiego ESP przez ESP-NOW.

Zasada działania:

* BMA400 w trybie low-power monitoruje wszystkie osie.
* Wykrycie ruchu ⇒ wake-up interrupt ⇒ przejście w tryb normal ⇒ INT1 = LOW.
* STM32 wybudzany z STANDBY przez pin PWR\_WAKEUP\_PIN1.
* STM32 co wybudzenie odczytuje FIFO z BMA400. Dzieli 32 próbki z osi Y na 4 bloki po 8. Dla każdego bloku liczy odchylenie standardowe. Jeżeli różnica odchylenia między sąsiednimi blokami przekracza 5 LSB (~5 mg), to uznaje to za oznakę ruchu bramy. Detekcja wykonywana jest cyklicznie przez cały czas, gdy FIFO dostarcza dane – czyli przez cały czas ruchu.
* Podczas wykrytego ruchu STM32 jest w stanie aktywnym. TIM14 przepełnia się co 1 ms. Gdy wartość czasu osiągnie wielokrotność 2 s, STM32 wysyła aktualny czas do ESP przez UART.
* Gdy przestanie wykrywać ruch (brama zatrzyma się), STM32 pobiera bieżący czas otwarcia i wysyła go do ESP. Potem przechodzi w tryb standby – wybudzenie następuje przez przerwanie z BMA400.