

# UIFlow – GPIO – IO

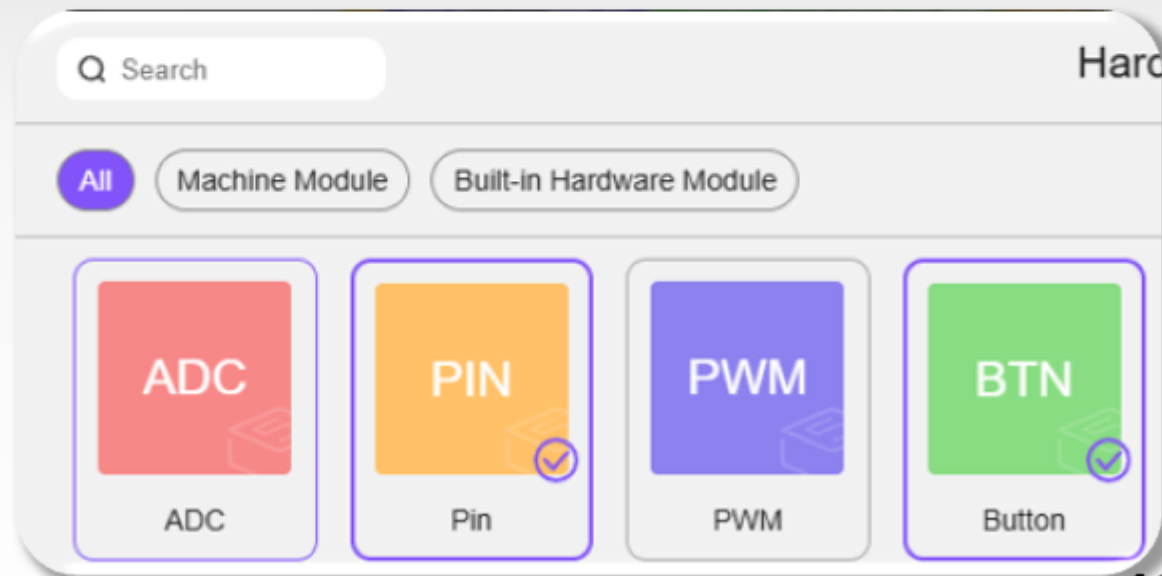
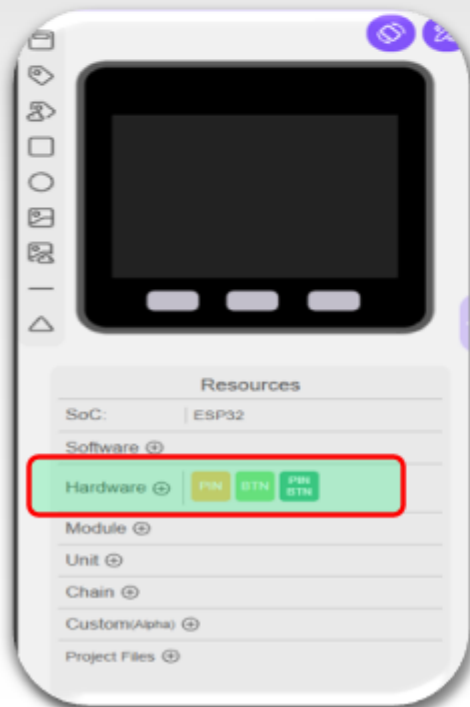
- **Porty cyfrowe – stan 0 lub 1, TRUE lub FALSE**
- **Pracują w trybie IN lub OUT**
- **Część portów GPIO jest bezpośrednio podpięta do elementów wewnętrznych mikrokontrolera:**
  - **Przyciski M5Stack CORE – GPIO39, GPIO38 i GPIO37**

<b>5V</b>	UART0		ISP		ADC			DAC		<b>GND</b>			
	1T	3R	EN	G0	34	35	36	25	26				
<b>3V3</b>	17T	16R	21D	22C	23	19	18	2	5	12	13	15	BAT
	UART2		I2C		MO	MI	SCK	GPIO					



# UIFLOW – GPIO – IO

## Grupa HARDWARE

- > **PIN** – porty cyfrowych
- > **BUTTON** – zintegrowane przyciski (A, B, C)
- > **PIN BUTTON** – przycisk zewnętrzny podpięty do portu cyfrowego



# UIFlow – PIN

**Pin**  

Init

Init Pin  mode  pull

Read/Get

get  value

Write/Set

on

off

Set  value

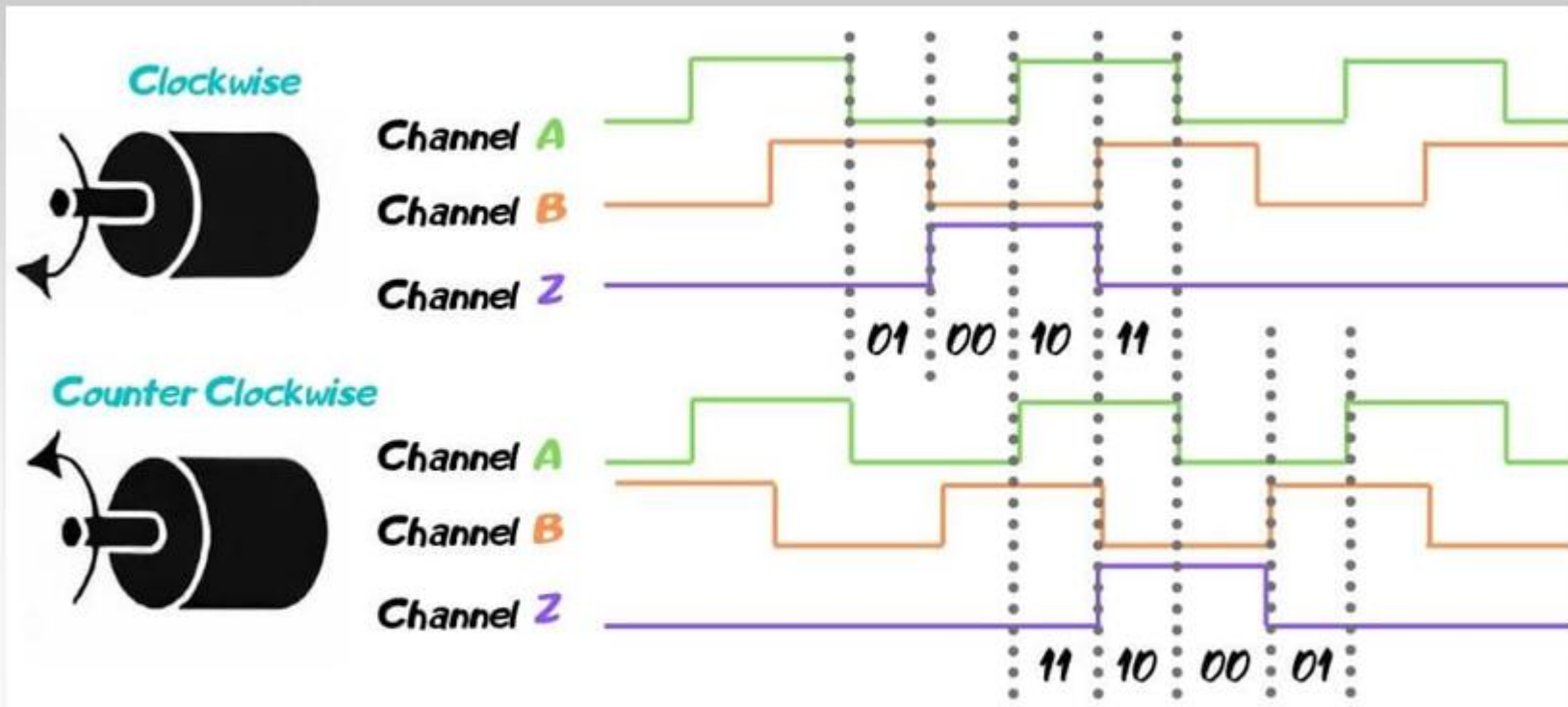
Set  value  (0 or 1)

Pin unused

**Pin**

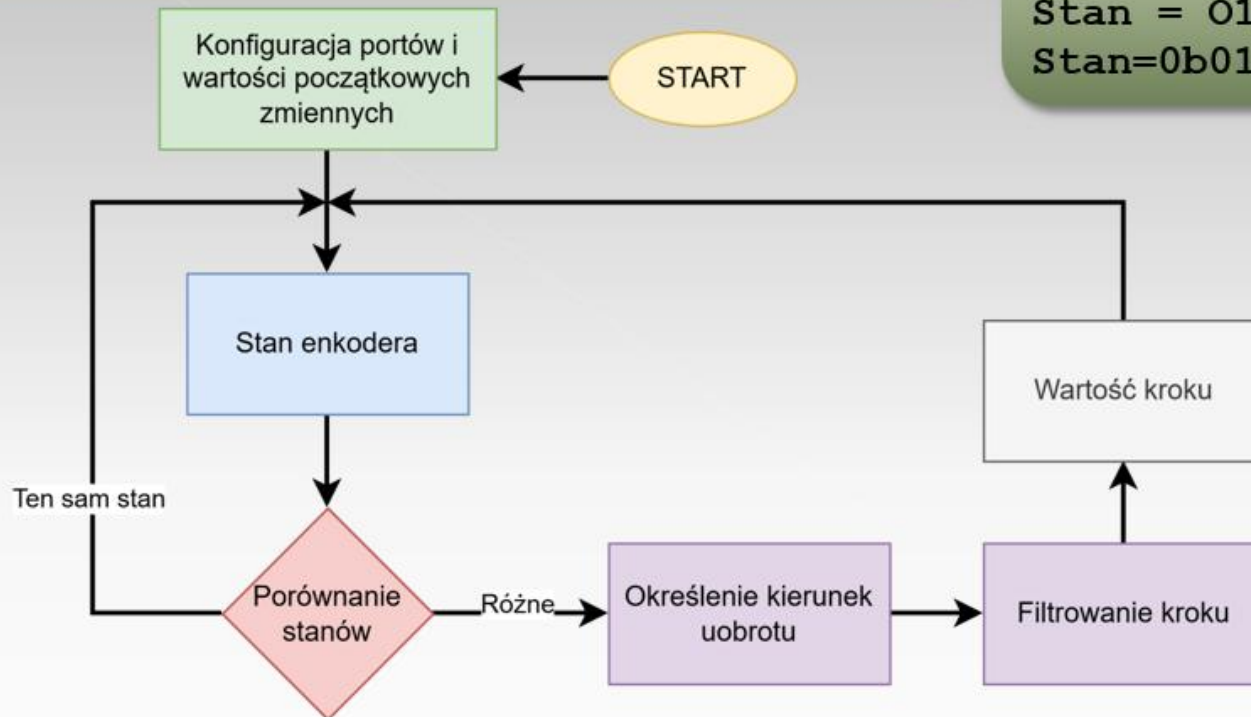
- Variables
- Math
- Loops
- Logic
- Functions
- Text
- Lists
- Tuple
- Map
- JSON
- Bytearray
- Bytes
- Hardware

# ENKODER



00 → 01 → 11 → 10 → 00 (kierunek +)  
 00 → 10 → 11 → 01 → 00 (kierunek -)

A=0 B=1  
 $O = A \ll 1 | B \rightarrow O = (AB) = 01$   
 $O1=01 \quad O2=11$   
 $Stan = O1 \ll 2 | O2 \rightarrow Stan = \{O1O2\} = 0111$   
 $Stan = 0b0111 = 1+2+4 = 7$



stany = [0, -1, +1, 0, +1, 0, 0, -1, -1, 0, 0, +1, 0, +1, -1, 0]

## ⊙ Zmienne

- > pozycja=0
- > poprzedni, aktualny
- > acc=0
- > tabela = [ 0, -1, +1, 0,  
                  +1, 0, 0, -1,  
                  -1, 0, 0, +1,  
                  0, +1, -1, 0 ]

## ⊙ Setup

- > poprzedni = (A << 1) | B

## ⊙ Loop

- **Zmienne**

- **Setup**

- **Loop**

- > aktualny= (A<<1) | B

- > **Sprawdzenie czy zmienił się stan**

- > index= (poprzedni<<2) | aktualny

- > acc+tabela[index]

- > poprzedni=aktualny

- > **Filtracja**

- **Jeżeli** acc>=4 **to** pozycja++ **i** acc=0

- **Jeżeli** acc<=-4 **to** pozycja-- **i** acc=0

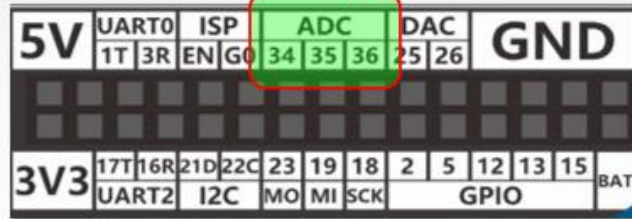
# UIFlow – GPIO – analog

## Przetwornik ADC

> 34, 35, 36

## PWM

> Każdy port cyfrowy GPIO



ADC

Init

Init Pin 34 attenuation 11DB(0 - 3.3V)

Read/Get

read Please init a ADC

read Please init a ADC u16

read Please init a ADC microvolts

Write/Set

Set Please init a ADC atten 0 - 1.2V

Set Please init a ADC width 0 - 511

Set ADC34 width 0 - 511

Set ADC34 atten 0 - 1.2V

PWM

Init

Init Pin 1 freq 20000 Hz (1 - 40000000) duty 512 (0 - 1023)

Please init a PWM dcint

Read/Get

get Please init a PWM duty

get Please init a PWM duty u16

get Please init a PWM freq

Write/Set

Set Please init a PWM duty 512 (0 - 1023)

Set Please init a PWM duty u16 32768 (0 - 65536)

Set Please init a PWM freq 30000 Hz (1 - 40000000)

- **Rozdzielczość przetwornika określa liczbę dyskretnych wartości jakie może on wytworzyć – wyrażaną w bitach.**

○ 8 <sub>bit</sub>	$2^8 = 256$	19.5	<u>mV</u> ( <u>V<sub>cc</sub></u> 5V)
○ 10 <sub>bit</sub>	$2^{10} = 1\ 024$	4.9	<u>mV</u>
○ 12 <sub>bit</sub>	$2^{12} = 4\ 096$	1.2	<u>mV</u>
○ 16 <sub>bit</sub>	$2^{16} = 65\ 536$	0.08	<u>mV</u>
○ 24 <sub>bit</sub>	$2^{24} = 16\ 777\ 216$	0.0003	<u>mV</u>

- **Przetwarzanie A/C tworzą 3 etapy: próbkowanie, kwantyzacja i kodowanie.**

# ADC

**Przetwarzanie ciągłego sygnału analogowego na sygnał cyfrowy polega na dyskretyzacji sygnału w czasie:**

1. próbkowaniu - pobieranie kolejnych próbek wartości sygnału w pewnych odstępach czasu, w taki sposób, aby ich ciąg umożliwił jak najwierniejsze odtworzenie całego przebiegu funkcji.
2. dyskretyzacji wartości sygnału czyli kwantowaniu - przyporządkowaniu każdej próbce skończonej liczby poziomów amplitudy, odpowiadającym dyskretnym wartościom od zera do pełnego zakresu.
3. kodowaniu uzyskanego sygnału dyskretnego.

180

