GPIO / DIO / IO

GPIO is a digital Module, responsible on control all the PINS on the SOC

It is called

  GPIO : General Purpose Input/Output

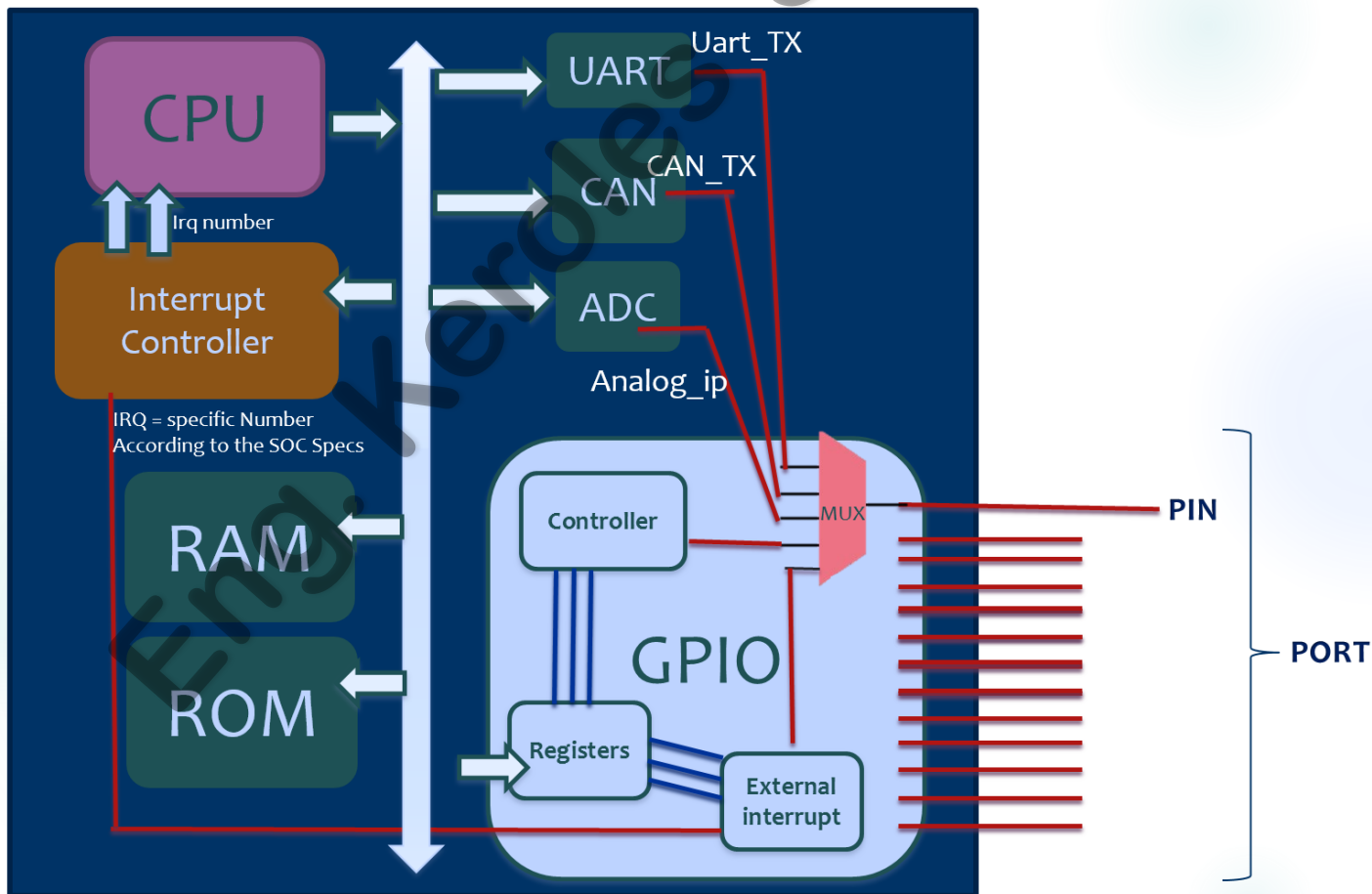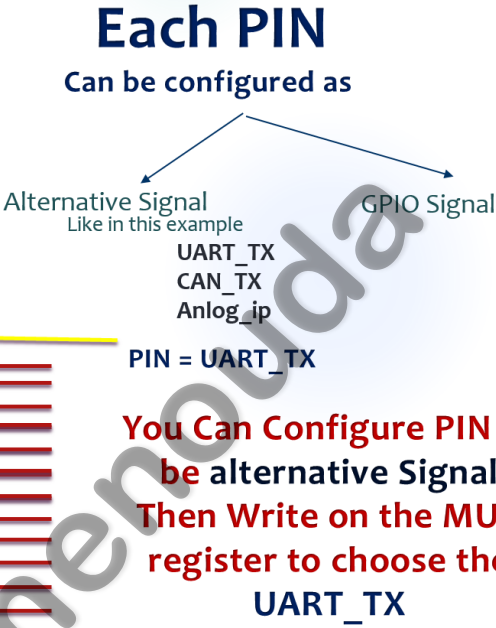 DIO : Directive input/Output

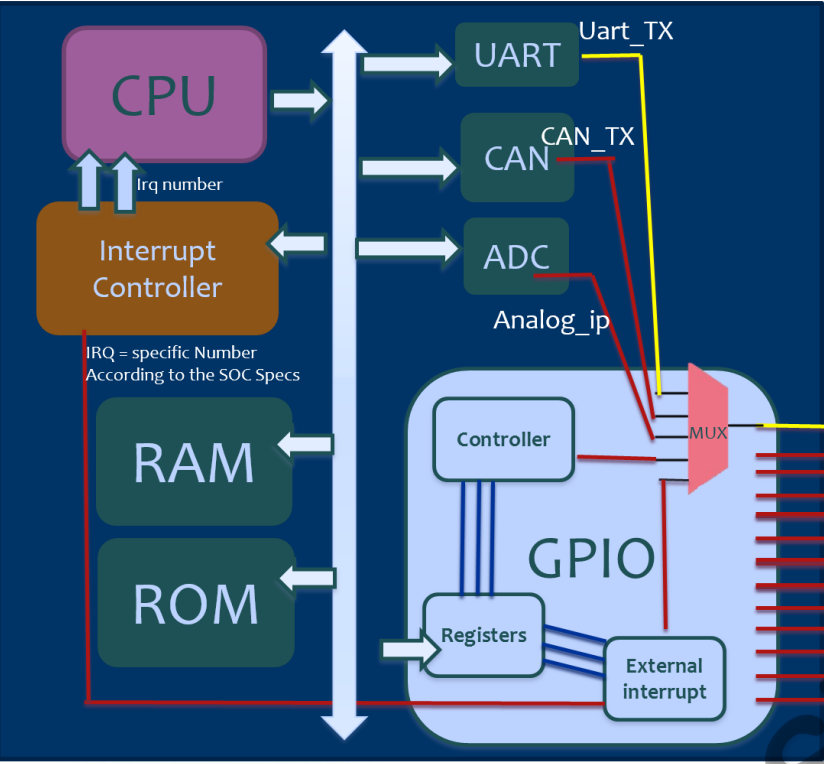 IO : input/Output

**I will describe the GPIO on a General Concept
and general Capabilities And Features.
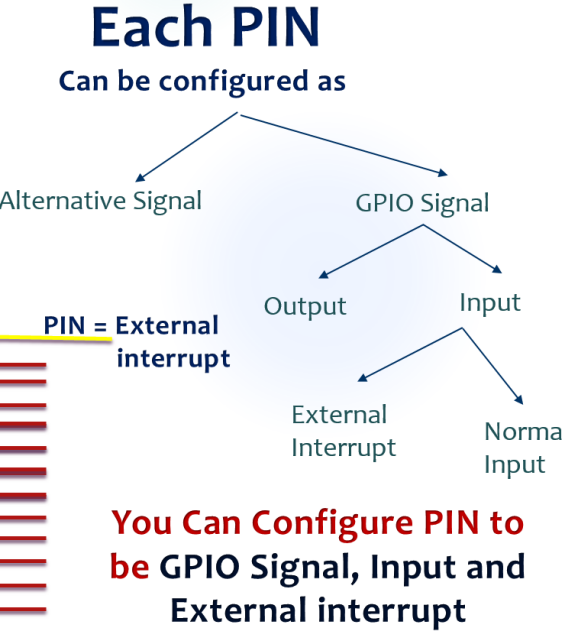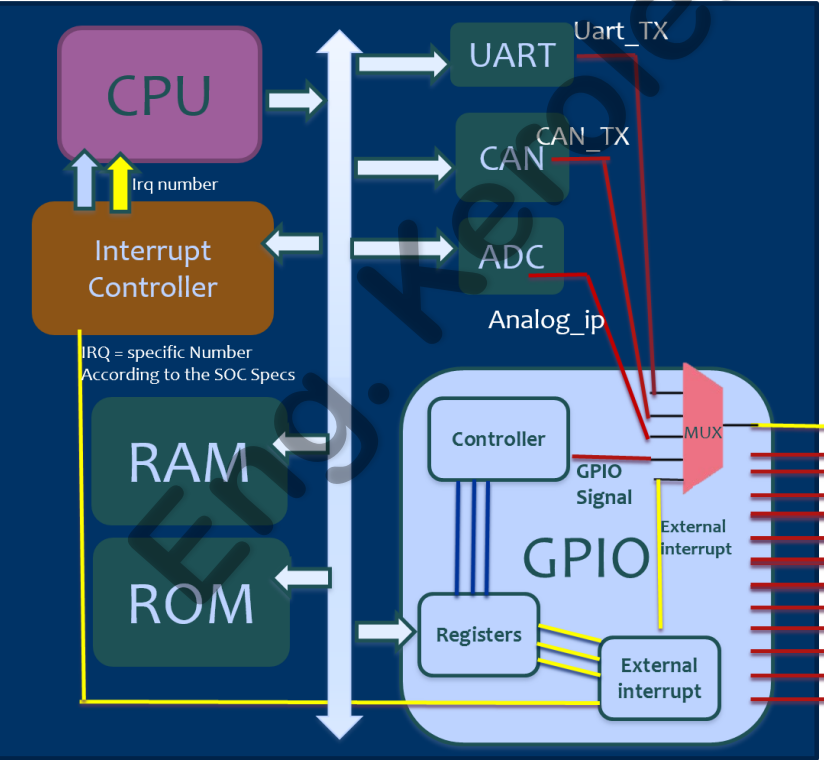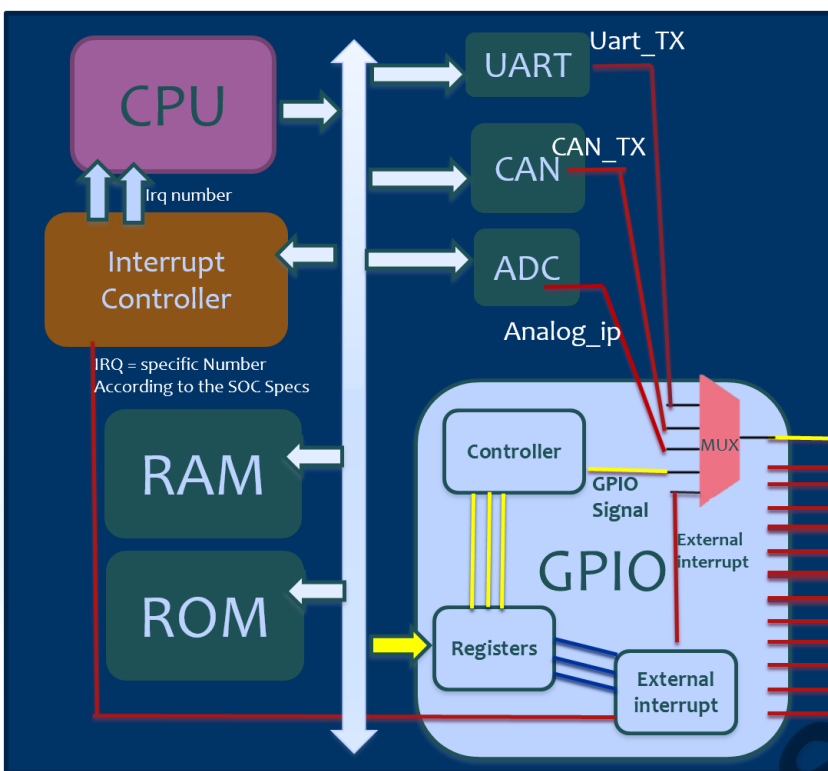So I created the following example as a case-study
Then will read the Atmega32 GPIO.
And on the future you will read the TivaC GPIO by yourself**

Diagram 1 (top)

CPU

Irq number

Interrupt Controller

IRQ = specific Number According to the SOC Specs

RAM

ROM

UART — Uart_TX

CAN — CAN_TX

ADC

Analog_ip

Controller

GPIO

Registers

External interrupt

MUX

Each PIN
Can be configured as

Alternative Signal
Like in this example
UART_TX
CAN_TX
Anlog_ip

GPIO Signal

PIN = UART_TX

You Can Configure PIN to be alternative Signal Then Write on the MUX register to choose the UART_TX

You can do the same for other alternative signals (CAN, ADC)

Diagram 2 (bottom)

CPU

Irq number

Interrupt Controller

IRQ = specific Number According to the SOC Specs

RAM

ROM

UART — Uart_TX

CAN — CAN_TX

ADC

Analog_ip

Controller

GPIO Signal

External interrupt

GPIO

Registers

External interrupt

MUX

Each PIN
Can be configured as

Alternative Signal

GPIO Signal

Output

Input

External Interrupt

Norma Input

PIN = External interrupt

You Can Configure PIN to be GPIO Signal, Input and External interrupt

ENG. Keroles Shenouda
https://www.facebook.com/groups/embedded.system.KS/

2

Top diagram

CPU

Irq number

Interrupt Controller

IRQ = specific Number According to the SOC Specs

RAM

ROM

UART — Uart_TX

CAN — CAN_TX

ADC

Analog_ip

Controller

GPIO Signal

External interrupt

Registers

GPIO

External interrupt

MUX

Each PIN

Can be configured as

Alternative Signal   GPIO Signal

Output   Input

PIN = INPUT

External Interrupt   Normal Input

External Interrupt   Normal Input

You Can Configure PIN to be
GPIO Signal, Normal Input
The CPU will read the GPIO
Data Register

Bottom diagram

CPU

Irq number

Interrupt Controller

IRQ = specific Number According to the SOC Specs

RAM

ROM

UART — Uart_TX

CAN — CAN_TX

ADC

Analog_ip

Controller

GPIO Signal

External interrupt

Registers

GPIO

External interrupt

MUX

Each PIN

Can be configured as

Alternative Signal   GPIO Signal

Output   Input

PIN = OUTPUT

External Interrupt   Normal Input
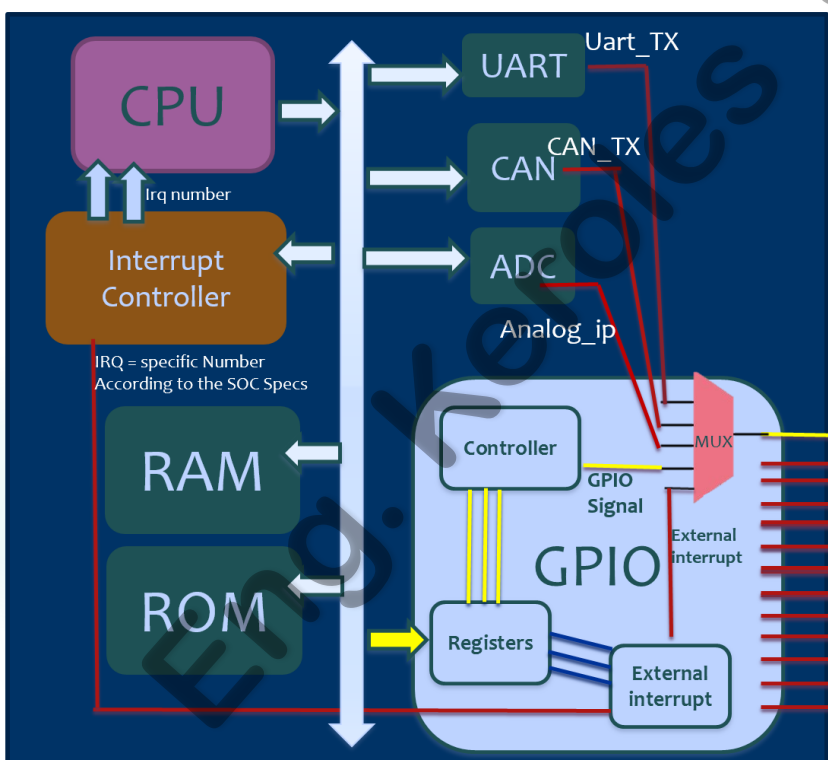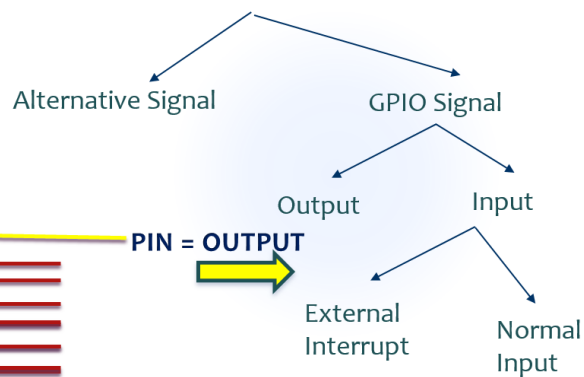
You Can Configure PIN to be
GPIO Signal, Output
The CPU will write the GPIO
Data Register

DDRXn at atmega32

```
                        DDRXn   at  atmega32
                              |
        ↓                                              ↓
        1                                              0
    output                                          input
                                                       |
                                        ↓                        ↓
                                    Pull up ?!            pull down ?!
so then you should              → "PORTxn" == 1
use   "PORTxn" Register         → "SFIOR" == 1<<2
                                   -ve logic
        ↓                    ↓
    PORTxn=1            PORTxn=0
```
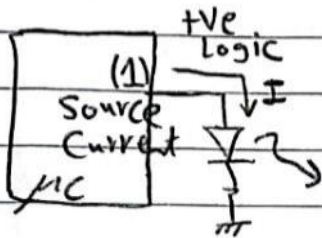
PORTxn=1 — (1) Source Current, +ve Logic, µC

PORTxn=0 — +ve Logic, µC, Sink Current, 0 VCC
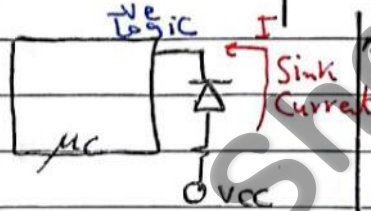
Pull up: "PORTxn" == 1 && "SFIOR" == 1<<2, -ve logic, 0 VCC, PUR, Source Current, µC
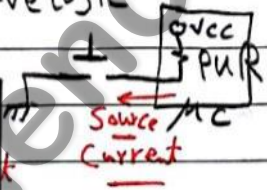
then you should to Read $PIN_{xn} == 0$ during pressing the Button

Hi z No current, PDR, µC, VCC
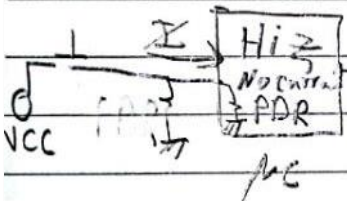
pull down happen if

$$(PORT_{xn} == 1 \ \&\& \ SFIOR == 1<<2)$$

$$|| \ OR$$

$$(PORT_{xn} == 0 \ \&\& \ SFIOR == 1<<2)$$