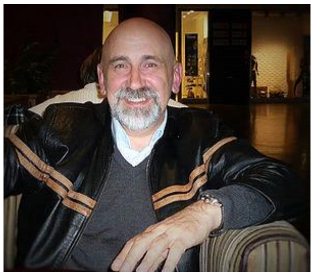
electronics-lis



https://electronics-lis.com/
info@electronics-lis.com

1.Few comments for all examples:

a. The main idea is to share very simple, commented and complete examples.
b. The PIC32MK is applying   an external HR clock 10 MHz.
c. All tests are based on PIC32MK1024MCF064 but they could be easy adapted for different devices
d. Pay attention on the UART installation: pins initialization and   channel number (printf).
e. The UART and its echo **(included into all examples)** is applying to communicate with test and verify if the PIC is working.
f. Some test are in /*   */ (example UART)
g. If you have any comments, improvements, please lets help us
h. The ADC test does'nt apply the "REGbits" solution.

2. PIC32MK-DMA-R02_ReceiverTest:

a. **Short description**
That is "byte by byte" UART receiver based on the DMA access. It is applying to interface an internal Command Interpreter. The example could be addapted for diffrent applications.
b. **Authors**
L.Lisowski, revision 01
L.Lisowski, revision 02
c. History
R01 => L.Lisowski , initial version,18-Jan-2018
R02 => L.Lisowski , initial version, 24-Jan-2018 (BGR calculation, cleaning)

3. PIC32MK-DMA-R02_TransmitterTest:

a. **Short description**
That is message transmitter on the UART (terminal) when you hit any key. I'm applying it to interface the user sending stored strings by DMA transmitter. Observe that message is set only to defined character.
b. **Autors**
L.Lisowski, revision 01
L.Lisowski, revision 02

c. History
R01 => L.Lisowski , initial version,18-Jan-2018
R02 => L.Lisowski , initial version,24-Jan-2018 (BGR calculation, cleaning)

4. PIC32MK-SPI2-RO2_Test:

a. **Short description**
In this example any pushed key on the keyboard is sent by SPI. You need to adapt pins and SPI's module number for your application. I'm using it to control 16 digital I/O implemented on the applied CPLD (internal shifter and latch) (see pdf). You can connected MISO and MOSI pins to implement a communication loop

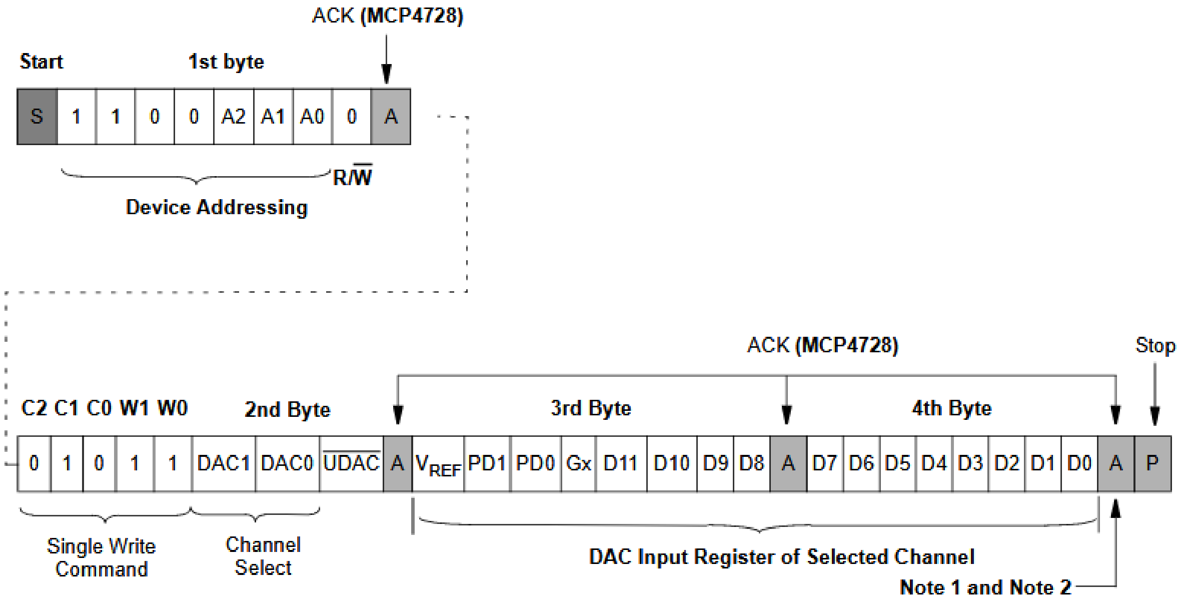
b. **Authors**
L.Lisowski, revision 01
L.Lisowski, revision 02

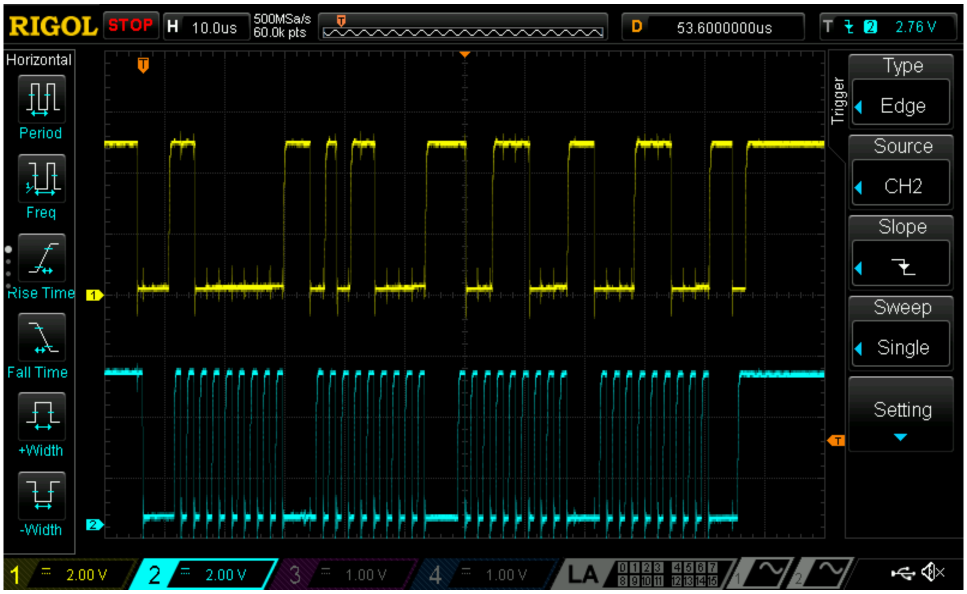
c. **History**
R01 => L.Lisowski , initial version,18-Jan-2018
R02 => L.Lisowski , initial version,24-Jan-2018 (BGR calculation, cleaning)

5. PIC32MK-I2C-RO1_Test:

a.**Short description**

That was certainly more complicated example. There are not any documentation (empty chapter) and any description. Finally, I've tested different channel and "discover" relation between one pair of pins and its number. I'm using this interface to configure my hardware after supply set-on. That for, I don't use **any interruption**. In any control loop only SPI should be applied (that is my opinion). The presented example interfaces excellent Microchip's 12 bits DAC MCP4728 (nonvolatile !!, small footprint, low power, different configurations and buffered outputs). It is applied into my applications to define shifting for internal ADC. The example includes also BRG calculation for I2C which are "acceptable" by XC32.
I've tried to be simpler **as possible**. This example can be extended to more complex procedures adding necessary **masks** and required sequences. It includes also while's loop **not blocking** implementation.





H 10.0us
500MSa/s
60.0k pts
D 53.6000000us
T 2 2.76 V
Type Edge
Source CH2
Slope
Sweep Single
1 2.00 V
2 2.00 V
3 1.00 V
4 1.00 V

a. **Autors**
L.Lisowski, revision 01

b. **History**
R01 => L.Lisowski , initial version, 24-Jan-2018

6. PIC32MK-ADC-RO2_Test :

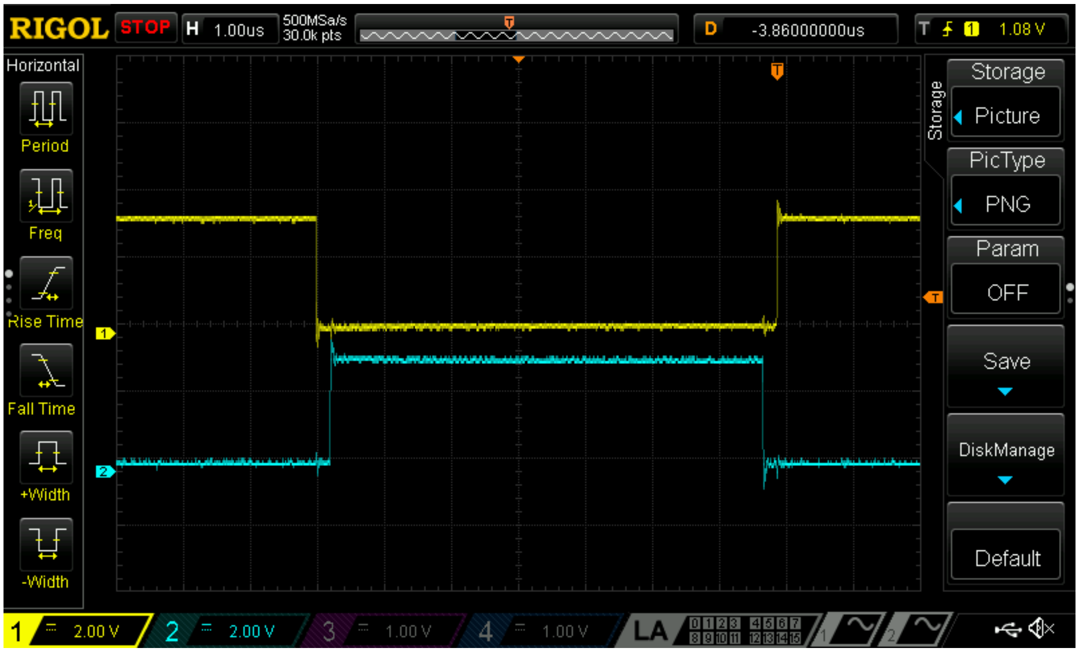
a. I'd essentially HW problems essentially related to interface pre-amplifiers. But ....The new example is ready and additionally I didn't applied "REGbits" solutions. The main idea is to initialize all 0 to 5 and 7 ADC converters except ANEN and digital bits in the general initialization. The applied converters should be enabled only for given application. Anyway, example is fully commented and includes few printf to control some parameters during tests. Enjoy you !

b. Authors
L.Lisowski R01 and R02

c. History
R01 => L.Lisowski , initial version, 20-Jan-2018 with HW problems
R02 => L.Lisowski , 07-feb-2018 cleaned version without REGbits solution

6. PIC32MK-PWM-RO1_Test :

a. That is very simple example for channel 1 as Complementary two pins PWM outpts. I've added Deat Time and its compensation. Enjoy you !



a. Authors
L.Lisowski R01

c. History
R01 => L.Lisowski , initial version, 28-Feb-2018

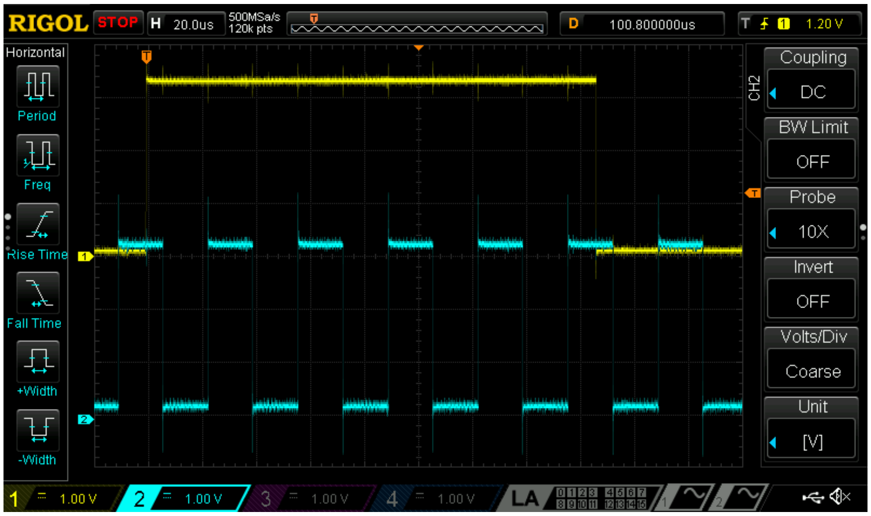
7. PIC32MK-PWM-INTERRUPTION-RO1_Test :

a. The main idea is to add ADC conversion in the middle of PWM step. To do it I've implemented the Trigger interruption service.

b. Additionally, to verify synchronization I've added two pins F0 and F1 to be connected with the oscilloscope.

c. Be careful: when you enable the PWM module
SETBIT(PTCON,15); //! PWM module enable)
the all unused PWM pins are overwritten (blocked). To have absolutely disable related PWM channels. For 64 pins packaging and F0 and F1:
PMD4SET = 0xFFF00000; //! Disable channels PWM5 to PWM12 !!!!!!!!!

d. That is very simple example for channel 1 as Complementary two pins PWM outpts. I've added also Death Time and its compensation. Enjoy you !

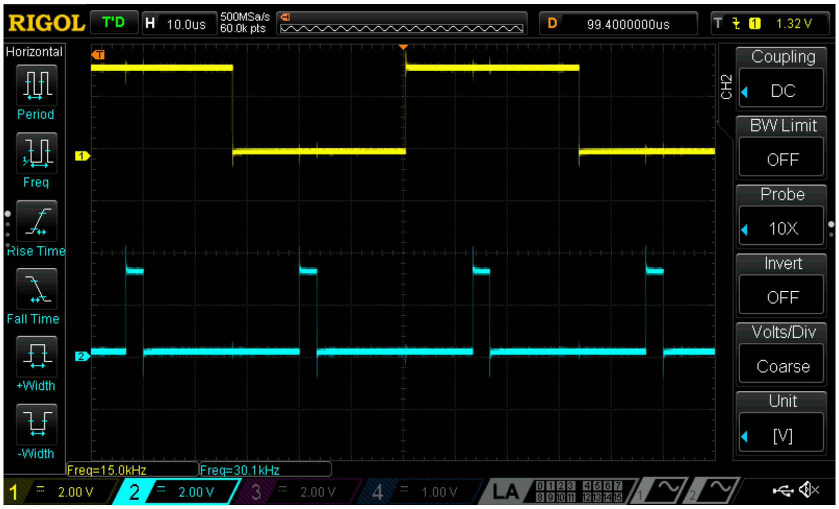

RIGOL STOP H 20.0us 500MSa/s 120k pts D 100.800000us T 1.20 V

Horizontal: Period, Freq, Rise Time, Fall Time, +Width, -Width

CH2
Coupling DC
BW Limit OFF
Probe 10X
Invert OFF
Volts/Div Coarse
Unit [V]

1 = 1.00 V 2 = 1.00 V 3 = 1.00 V 4 = 1.00 V LA

TRGDIV1=4, Top=> FO toggled, Bottom=> PWM (50%)


RIGOL T'D H 10.0us 500MSa/s 60.0k pts D 99.4000000us T 1.32 V

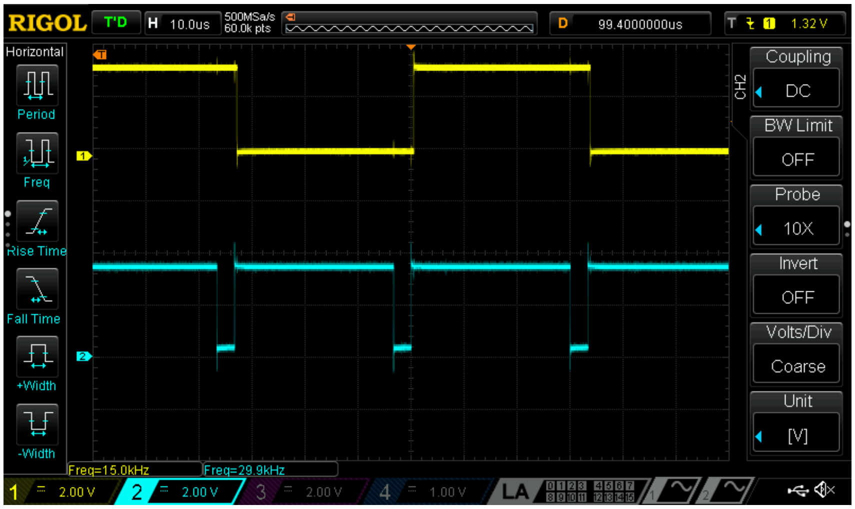
Horizontal: Period, Freq, Rise Time, Fall Time, +Width, -Width

CH2
Coupling DC
BW Limit OFF
Probe 10X
Invert OFF
Volts/Div Coarse
Unit [V]

Freq=15.0kHz Freq=30.1kHz

1 = 2.00 V 2 = 2.00 V 3 = 2.00 V 4 = 1.00 V LA

TRGDIV1=0, Top=> FO toggled, Bottom=> PWM (90%)


RIGOL T'D H 10.0us 500MSa/s 60.0k pts D 99.4000000us T 1.32 V

Horizontal: Period, Freq, Rise Time, Fall Time, +Width, -Width

CH2
Coupling DC
BW Limit OFF
Probe 10X
Invert OFF
Volts/Div Coarse
Unit [V]

Freq=15.0kHz Freq=29.9kHz

1 = 2.00 V 2 = 2.00 V 3 = 2.00 V 4 = 1.00 V LA

TRGDIV1=0, Top=> FO toggled, Bottom=> PWM (10%)

a. Authors
L.Lisowski R01

c. History
R01 => L.Lisowski , initial version, 16-Mar-2018