SYNCHRONISM UNIT 1PPS WITH LPC1768

Alberto Río Martínez, Adrián Alonso, Pablo García, Carlos Almendros, Francisco Valle

IT-CDT 2021-13

Observatorio de Yebes

Apdo. 148 19080 Guadalajara

SPAIN

Phone: +34 949 29 03 11

Fax: +34 949 29 00 63

REVISION HISTORY

Version	Date	Author	Updates
1.0	13/07/2021	Alberto Río Martínez	First Version

INDEX

1. Introduction

The following system is an improvement of OAY-11 Synchronism Unit with LPC1768 uController, for ending with original analog system errors. OAY-11 Synchronism Unit goal is the study of hydrogen maser stability by comparing with a GPS signal from Yebes Observatory´s VLBA, and independent from STM (Station Timing Module).

Figure 1 diagram shows the basic operation of the system: 5MHz input from maser is introduced by P1.28 pin, that counts rising edges from input signal and determines low and high level times (100us high, that is GPS signal´s level up time, and 999.9ms low, that is resting time) in P1.18 pin output by an internal clock from LPC1768 PWM register. GPS signal is introduced by P2.10 pin, that generates an interrupt every input´s rising edge. Reset happens if P2.10 interrupt is generated and reset button (connectetd to P1.26 pin) is pressed. As GPS input frequency is 1Hz, is advisable to press the reset button more than one second.

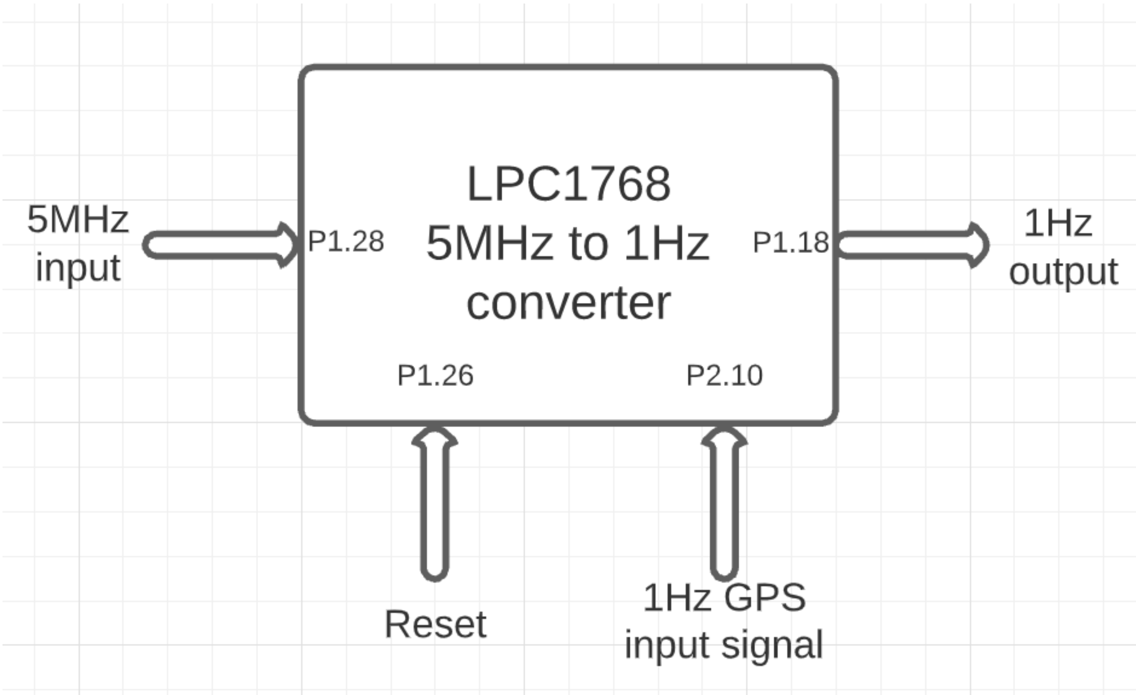

Figure 1: Block diagram of the system.

2. Equipment

2.1. LPC1768

LPC1768-Mini-DK2 is a small evaluation board production by Haoyu electronic, it based on the NXP (NXP Semiconductors) LPC1700 series processors (Cortex-M3 core).

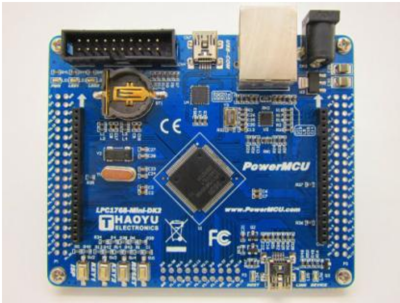

Figure 2: Top view of LPC1768 board.

The reason to use this board, is that Cortex-M3 processor has an appropiate speed (max. CPU speed of the board is 100MHz) for the 5MHz input signal. Other boards like raspberry pi can´t work with very high frequencies like this because it OS makes constant interrupts that slow down the CPU max. Speed, therefore this board lacks an OS. **Is important not exceed 3.3V GPIO input value.**

2.2. USB MINI-JTAG

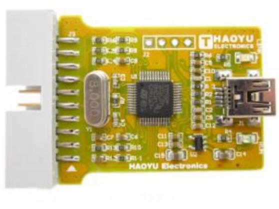

Figure 3: Top view of USB MINI-JTAG debugger.

USB-MiniJTAG is an USB powered JTAG emulator supporting a large number of CPU cores. It can supported by all major IDEs such as IAR EWARM, Keil MDK, Rowley CrossWorks, Atollic TrueSTUDIO, IAR EWRX, Renesas HEW, Renesas e2studio, and many others.

2.3. Keil uVision5

Keil µVision combines project management, run-time environment, build facilities, source code editing, and program debugging in a single environment. The program has specific libraries for LPC1768, and initialize and load the code on the board. With the debugger (USB MINI-JTAG)is possible to execute the code step by step.

3. Code explanation line by line

3.1. Initialization

-Lines 4 to 8: Functions declaration.

<u>Config_GPIO:</u>

-Lines 13 and 14: P1.26 and P2.10 definition as inputs setting to low the pins. As is shown in Figure 5 P1.26 is default in GPIO mode, this mode is used because P1.26 is the synchronization button.

```
1  #include <LPC17xx.h>
2  
3  //Declaración de las funciones
4  void initPWM(void);
5  void config_GPIO (void);
6  void inicioPINSEL(void);
7  void inicioIRQs(void);
8  void EINT0_IRQHandler(void);
9  
10 
11 void config_GPIO (void)
12 {
13   LPC_GPIO1->FIODIR &= ~(1<<26);  /* P1.26 definido como entrada */
14   LPC_GPIO2->FIODIR &= ~(1<<10);  // P2.10 definido como entrada
15 }
```

Figure 4: Initialization part of the code.

PINSEL3	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
21:20	P1.26	GPIO Port 1.26	MCOB1	PWM1.6	CAP0.0	00

Figure 5: P1.26 functions from UM10360, page 120.

3.2. EINT interrupt

The objective is to define the input of GPS signal as an interrupt input that is accessed every time a rising edge on GPS signal takes place. If during an interrupt, synchronize button is pressed, the system restarts, therefore synchronize button should be pressed more tan 1 second, that is the period of GPS signal. The explanation of the functions is:

inicioPINSEL:

-Line 20: P2.10 definition as EINT0 (external interrupt 0) setting to high pin 20 from PINSEL4, as is shown in Figure 7.

-Line 22: EINT0 configured as edge sensitive setting to high bit 0 of EXTMODE register, as is shown in Figure 9.

-Line 23: EINT0 configured as rising-edge sensitive setting to high bit 0 of EXTPOLAR register, as is shown in Figure 10 (every time a rising edge is read in P2.10 the interrupt is accessed).

inicioIRQs: Set priority and enable EINT0 interrupt.

EINT0_IRQHandler: This interrupt is accessed every time a rising edge is read in P2.10.

-Line 38: Clear the interrupt flag from EINT0 setting to high bit 0 of EXTINT register, as is shown in Figure 8.

-Line 39: Check that P1.26 has a low level (synchronize button is pressed).

-Line 40: If P1.26 is pressed, initialize PWM function.

```
17 void inicioPINSEL(void)
18 {
19 //Configuración del pin P2.10 como entrada de interrupción, por este pin se introducirá la señal de GPS de 1Hz
20  LPC_PINCON->PINSEL4 |= 1 << (10*2);   // P2.10 como EINT0
21
22  LPC_SC->EXTMODE |= (1<<0);  // configuramos EINT0 activo por flanco
23  LPC_SC->EXTPOLAR |= (1<<0);   //de subida (en todos los flancos de subida de la señal de GPS se accederá a la interrupción EINT0)
24 }
25
26 void inicioIRQs(void)
27 {
28 //Asignación de prioridades, esto se puede omitir, ya que solo hay una única interrupción
29   NVIC_SetPriorityGrouping(4);
30   NVIC_SetPriority(EINT0_IRQn, 0x2);
31
32 //Habilitación de la interrupción
33   NVIC_EnableIRQ(EINT0_IRQn);
34 }
35
36 void EINT0_IRQHandler(void)
37 {
38   LPC_SC->EXTINT |= (1);   // Borrar el flag de la EINT0 --> EXTINT.0
39   if (((LPC_GPIO1->FIOPIN>>26) & 0x01)==0){
40     initPWM(); //Si el pin p1.26 está a nivel bajo se ejecuta la funcion initPWM
41   }
42 }
```

Figure 6: EINT configuration and intialization.

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
21:20	P2.10	GPIO Port 2.10	EINT0	NMI	Reserved	00

Figure 7: P2.10 functions from UM10360, page 120.

Bit	Symbol	Description	Reset value
0	EINT0	In level-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0

Figure 8: EXTINT definition from UM10360, page 27.

Bit	Symbol	Value	Description	Reset value
0	EXTMODE0	0	Level-sensitivity is selected for EINT0.	0
		1	EINT0 is edge sensitive.	

Figure 9: EXTMODE definition from UM10360, page 28.

Bit	Symbol	Value	Description	Reset value
0	EXTPOLAR0	0	EINT0 is low-active or falling-edge sensitive (depending on EXTMODE0).	0
		1	EINT0 is high-active or rising-edge sensitive (depending on EXTMODE0).	

Figure 10: EXTPOLAR definition from UM10360, page 28.

3.3. PWM Register

PWM Register introduces 5MHz input from maser by P1.28 pin, that counts rising edges from input signal and determines low and high level times (100us high, 999.9ms low) in P1.18 pin output. The code lines explain is:

-Line 46: Select PWM function with PCONP register.

-Line 48: P1.18 definition as PWM1.1 (PWM output) setting to high pin 5 from PINSEL3, as is shown in Figure 12.

-Line 49: P1.28 definition as PCAP1.0 (PWM input) setting to high pin 25 from PINSEL3, as is shown in Figure 12.

-Line 51: PCR register is used to enable and select the type of each PWM channel. The selection is Single Edge PWM by setting to low pins 0,1 from PCR register (the selection is single edge by default, so this line can be deleted), as is shown in Figure 16.

-Line 52: PWM1 configured as PCAP1.0´s edge sensitive setting to high bit 0 of PWM1CTCR register, as is shown in Figure 14.

-Lines 53 and 54: MR1 selected at edge count value equivalent to 100us, and MR0 selected at edge count value equivalent to 1s. MR1 selects up level time, and MR0 the period of output signal. MR Register is shown in Figure 15.

-Line 55: MCR register reset edge count when MR0 happens (so reset at the end of every cycle for generate a new cycle again) by setting to high bit 1 from MCR register.

-Line 57: Enable PWM1.1 output by setting to high bit 9 from PCR register, as is shown in Figure 16.

-Lines 58 and 60: Enable PWM mode and reset TC after cycle count for generate new cycle by setting to high bits 0 and 1 from TCR register, as is shown in Figure 13.

```
44  void initPWM(void)
45  {
46    LPC_SC->PCONP |= 1 << 6; //seleccionar funcion PWM
47    LPC_PWM1->TCR = (1<<1); //Reset del TC y PR de PWM1
48    LPC_PINCON->PINSEL3 |= (1<<5); //P1.18 como salida PWM1.1
49    LPC_PINCON->PINSEL3 |= (0<<24) | (1<<25); //PCAP1.0 para pin p1.28
50
51    LPC_PWM1->PCR = 0x0; //Selección de Single Edge PWM
52    LPC_PWM1->CTCR = (0<<1) | (1<<0); //CONTAR EN FLANCOS DE SUBIDA de pcap1.0
53    LPC_PWM1->MR0 = 5000000; //Match0 cuando pase un segundo
54    LPC_PWM1->MR1 = 5000000/10000; //Match1 cuando pasen 100us
55    LPC_PWM1->MCR = (1<<1); //Reset del TC de PWM cuando se alcanza el match 0
56    LPC_PWM1->LER = (1<<1) | (1<<0); //guardar valores en MR0 Y MR1
57    LPC_PWM1->PCR = (1<<9); //activar salida PWM1.1
58    LPC_PWM1->TCR = (1<<1); //Reset del TC y PR de PWM1
59
60    LPC_PWM1->TCR = (1<<0) | (1<<3); //Activar contadores y modo PWM
61  }
```

Figure 11: PWM initialization.

PINSEL3	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
5:4	P1.18	GPIO Port 1.18	USB_UP_LED	PWM1.1	CAP1.0	00
25:24	P1.28	GPIO Port 1.28	MCOA2	PCAP1.0	MAT0.0	00

Figure 12: P1.28 and P1.18 functions from UM10360, pages 119, 120.

Bit	Symbol	Value	Description	Reset Value
0	Counter Enable	1	The PWM Timer Counter and PWM Prescale Counter are enabled for counting.	0
		0	The counters are disabled.	
1	Counter Reset	1	The PWM Timer Counter and the PWM Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until this bit is returned to zero.	0
		0	Clear reset.	

Figure 13: PWM1TCR definition from UM10360, page 527.

Bit	Symbol	Value	Description	Reset Value
1:0	Counter/ Timer Mode	00	Timer Mode: the TC is incremented when the Prescale Counter matches the Prescale Register.	00
		01	Counter Mode: the TC is incremented on rising edges of the PCAP input selected by bits 3:2.	
		10	Counter Mode: the TC is incremented on falling edges of the PCAP input selected by bits 3:2.	
		11	Counter Mode: the TC is incremented on both edges of the PCAP input selected by bits 3:2.	
3:2	Count Input Select		When bits 1:0 of this register are not 00, these bits select which PCAP pin which carries the signal used to increment the TC.	00
		00	PCAP1.0	
		01	PCAP1.1 (Other combinations are reserved)	
31:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Figure 14: PWM1CTCR definition from UM10360, page 528.

Bit	Symbol	Valu e	Description	Reset Value
0	PWMMR0I	1	Interrupt on PWMMR0: an interrupt is generated when PWMMR0 matches the value in the PWMTC.	0
		0	This interrupt is disabled.	
1	PWMMR0R	1	Reset on PWMMR0: the PWMTC will be reset if PWMMR0 matches it.	0
		0	This feature is disabled.	
2	PWMMR0S	1	Stop on PWMMR0: the PWMTC and PWMPC will be stopped and PWMTCR[0] will be set to 0 if PWMMR0 matches the PWMTC.	0
		0	This feature is disabled	

Figure 15: PWM1MR definition from UM10360, page 528.

Table 452: PWM Control Register (PWM1PCR - address 0x4001 804C) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	Unused		Unused, always zero.	NA
9	PWMENA1	1	The PWM1 output enabled.	0
		0	The PWM1 output disabled.	

Figure 16: PWM1PCR definition from UM10360, page 530, 531.

3.4. Main

Main only call functions explained before.

```
int main(void)
{
   config_GPIO ();
   inicioPINSEL();
   inicioIRQs();

   while(1);
}
```

Figure 17: Main of the code.

4. Experimental results

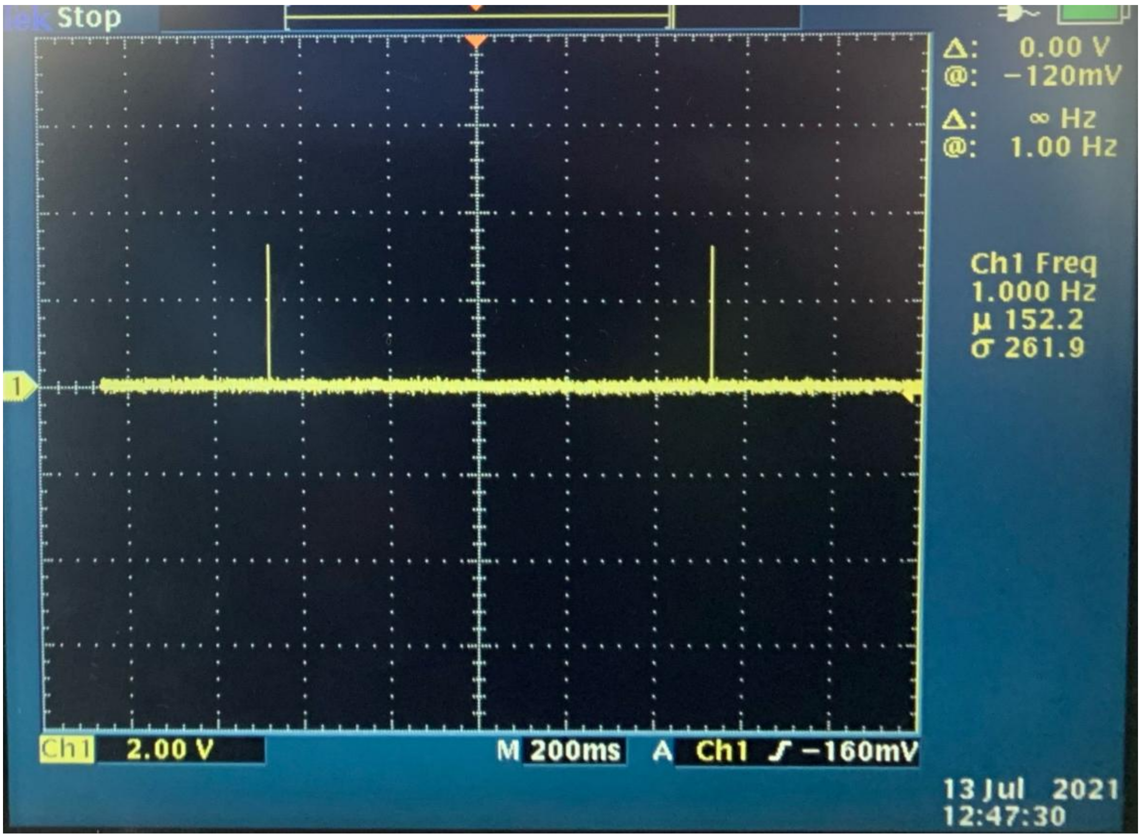

Figure 18: P1.18 pin output signal. Is shown the signal frequency of 1Hz and pulse width of 100us. The two square input signals are introduced by a signal generator, and have the frequency values of 5MHz (maser) and 1Hz (GPS), with amplitude of 1.5Vpp.

5. References

- UM10360 LPC176x/5x User manual - NXP

6. Appendix1: Keil uVision environment:

6.1. Create Project with Keil uVision5 for LPC1768 device

1. First of all, download Keil uVision5 and LPC17xx pack from Keil.com.

2. Pulse Project-New uVison Project and save it in a file previusly created.Then a window like Figure 19 one appears. We are going to select our device, pulse NXP-LPC1700 series-LPC176x-LPC1768 as is shown in Figure 20 and pulse OK.

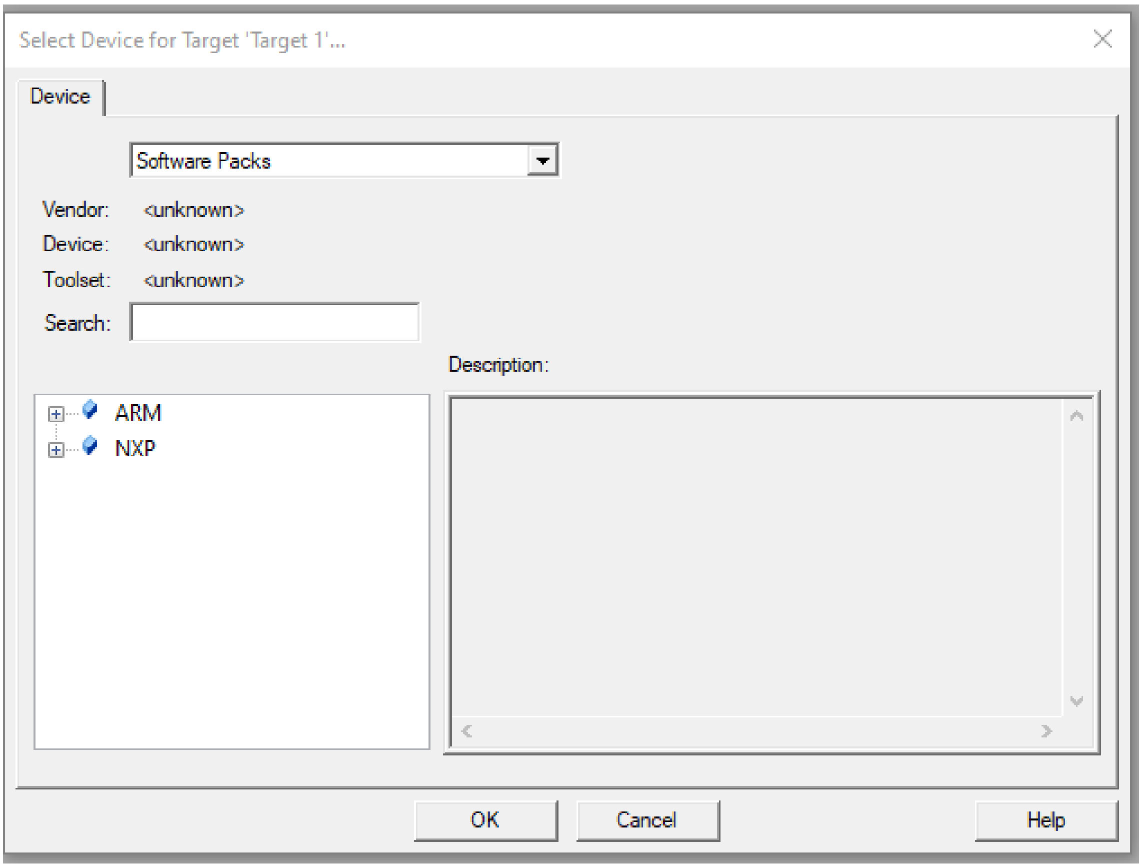

Figure 19

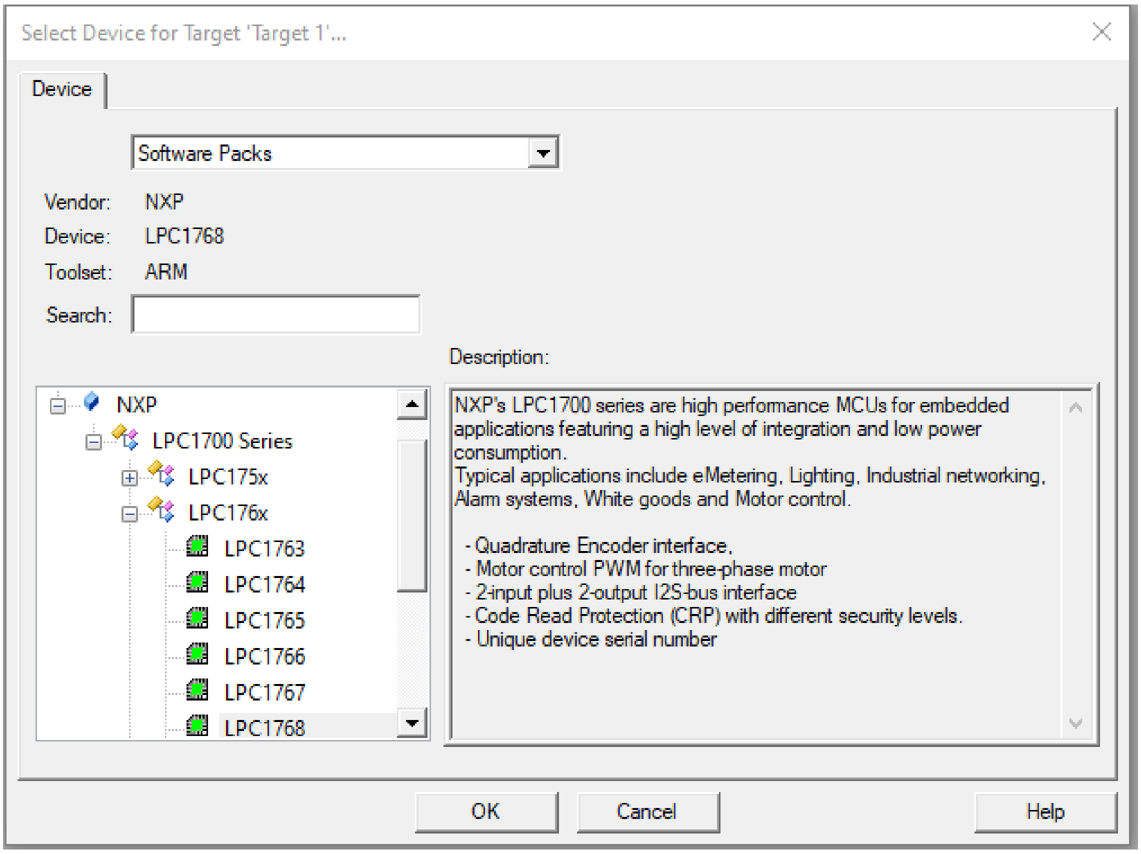
Select Device for Target 'Target 1'...
Device
Software Packs
Vendor: NXP
Device: LPC1768
Toolset: ARM
Search:
Description:
NXP
LPC1700 Series
LPC175x
LPC176x
LPC1763
LPC1764
LPC1765
LPC1766
LPC1767
LPC1768
NXP's LPC1700 series are high performance MCUs for embedded applications featuring a high level of integration and low power consumption.
Typical applications include eMetering, Lighting, Industrial networking, Alarm systems, White goods and Motor control.
- Quadrature Encoder interface,
- Motor control PWM for three-phase motor
- 2-input plus 2-output I2S-bus interface
- Code Read Protection (CRP) with different security levels.
- Unique device serial number
OK
Cancel
Help

Figure 20

3. Then a window like Figure 21 one appears, is required to activate the cells CMSIS-CORE and Device-GPIO-Startup like in Figure 22.

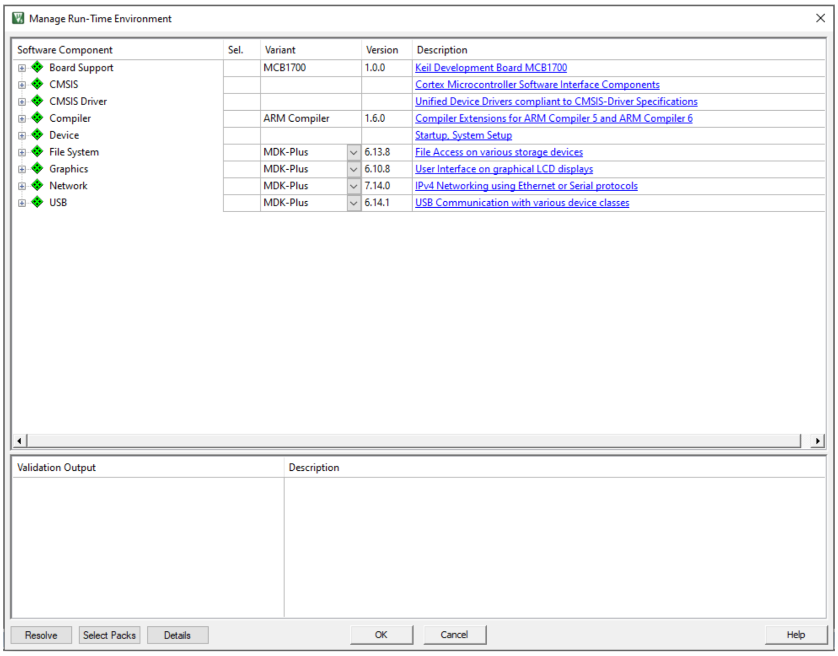

Figure 21

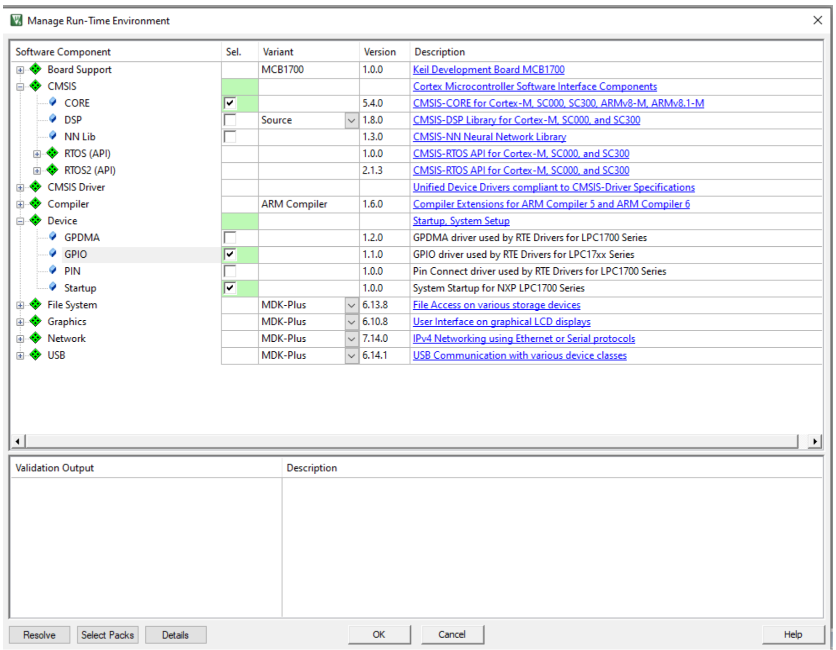

Figure 22

4. Then,a project is created with Figure 23 files. We are going to create a mian.c file: pulse Source Group 1 with right botton of the mouse and select Add new ítem to Group "Source Group 1". Then in the pop-up window select C File, write the name of the file and pulse Add as ism shown in Figure 24. For ending we can see our new file in Figure 25, you can write the code in main.c and load it in LPC1768.

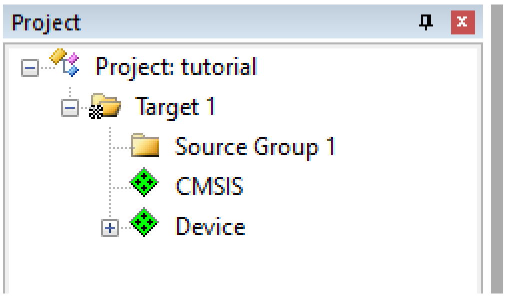

Figure 23

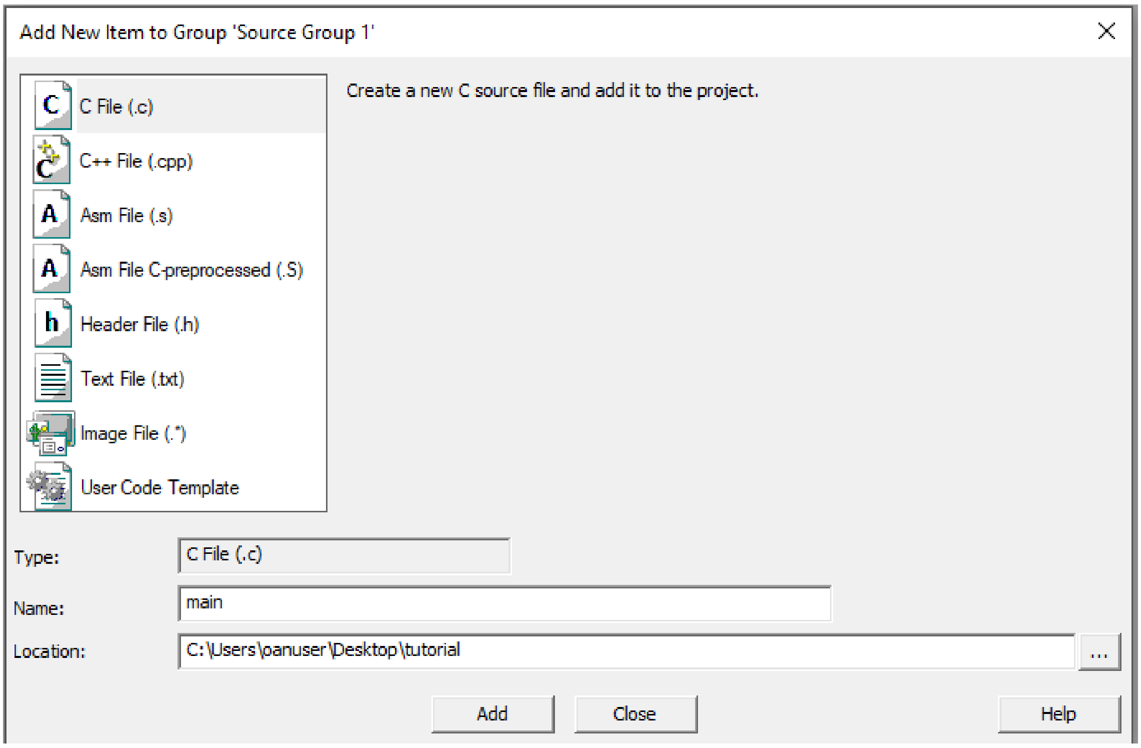

Figure 24

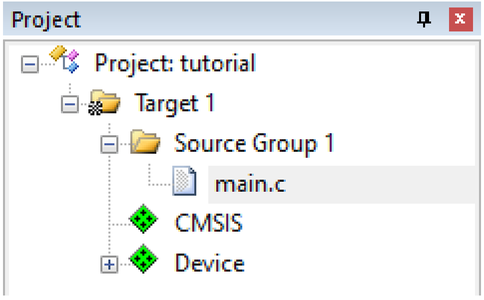

Figure 25

6.2. Drivers installation for USB MINI-JTAG Debugger use in uVision

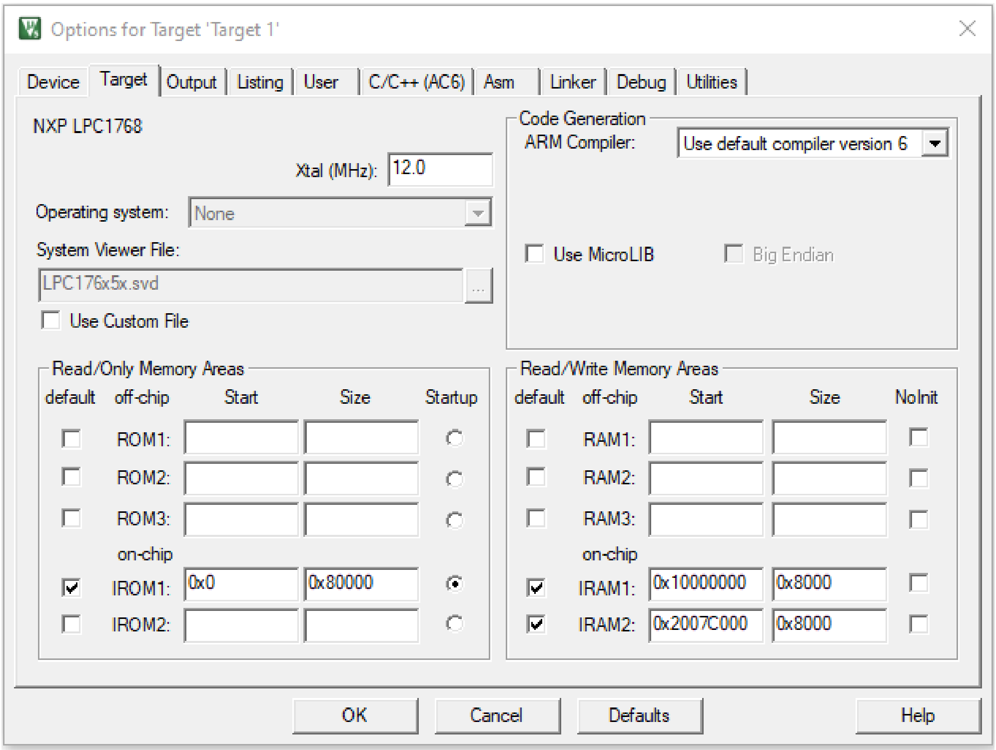

Figure 26: Pulse Target 1- Options for Target “Target 1”-Debug

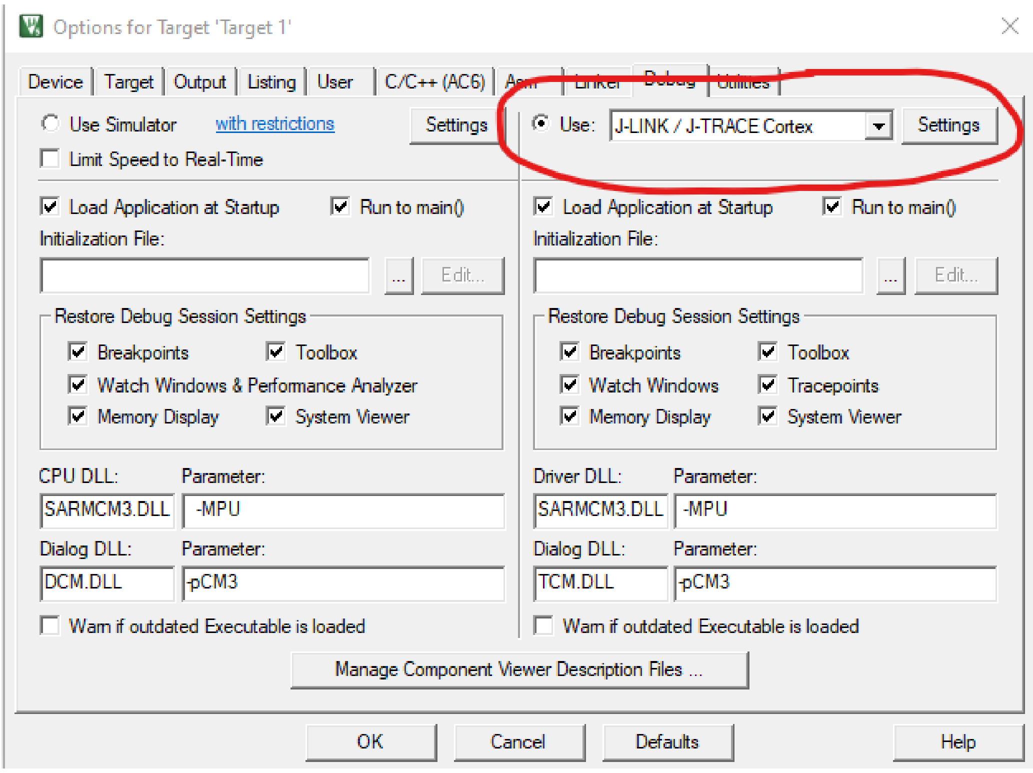

Figure 27: In debug, select J-LINK/J-TRACE Cortex in the rounded area for our USB-MINIJTAG and pulse Settings.

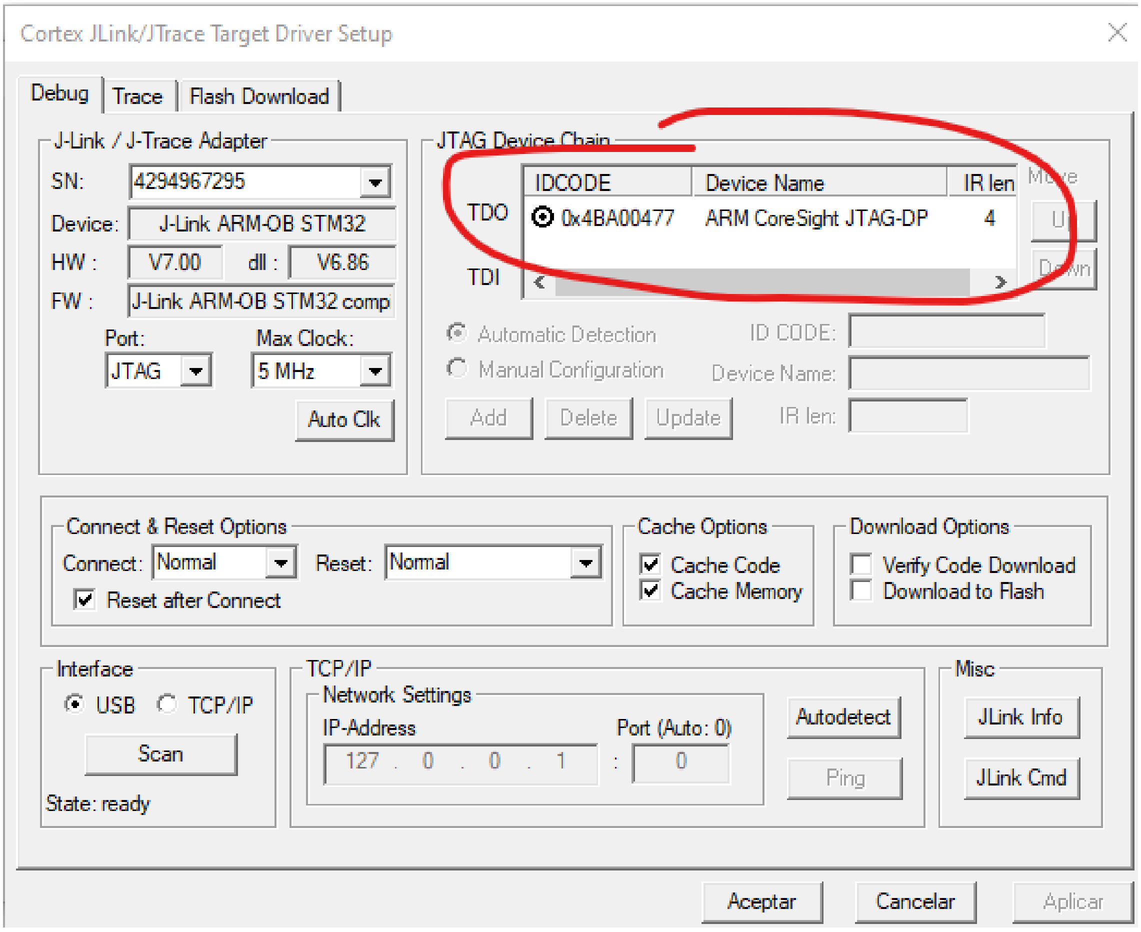

Figure 28: If in settings appear the text of the rounded area the JTAG has been selected fine, if not you must to install drivers from Figure 29.

Figure 29: Realtek high definition driver and J-LINK driver for correct selection of debugger.

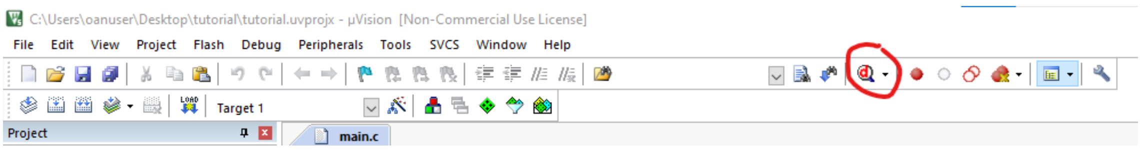

Figure 30: For execute the code with debugger pulse the button of rounded area.

7. Appendix2: input signal aconditioning board:

5MHz output signal from Hydrogen Maser has a 5V peak to peak voltage value. LPC1768 GPIO pins maximum voltaje is 3V3, and not accept negative voltages, so the input from the maser should be aconditioned. Design has been done in KiCad tool for it later printing on LPKF.

Schematic for signal aconditioning is shown in Figure 31, it consists on introducing Maser signal by Vin port, that is the non inverting input of LM360 comparator, and it compare Vin with ground from inverting input. LM360 is a comparator that compare twice input signals, when the non inverting input is higher than the inverting the output is set to high level, when is smaller the oputput is set to low. Is required to do a symmetric voltage supply. There are two 1uF capacitors for supply voltage, one 10nF capacitor for filtering at the input, a 50 ohms resistance in parallel at the input to match impedances with Maser signal, and a resistive divisor at the output that decreases comparator output voltage by half. In the schematic there are, too, a pull up 5Kohms resistor with a pulse connected to P1.26 pin from LPC1768, that is the push button to synchronize the system.

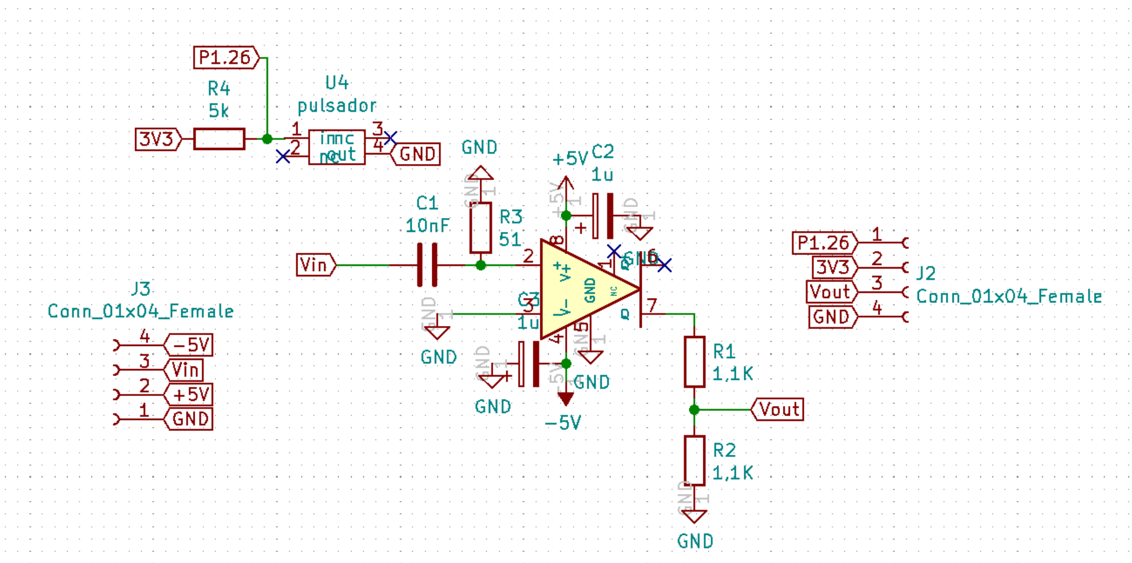

Figure 31: Schematic.

Figures 32 and 33 show the routing of the PCB and Figure 34 is the real PCB printed with LPKF.

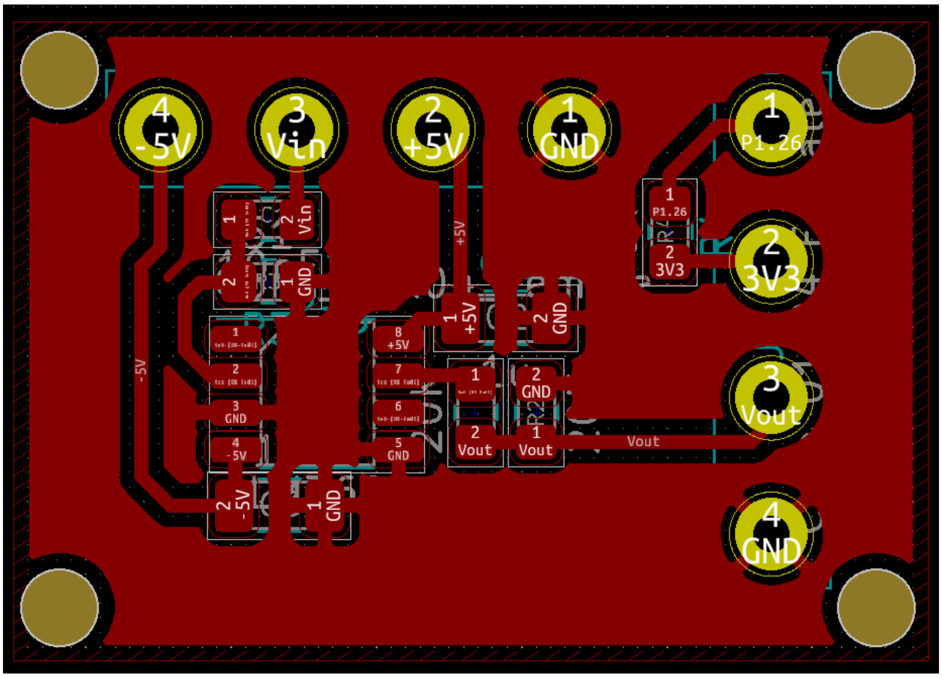

Figure 32: PCB with ground plane.

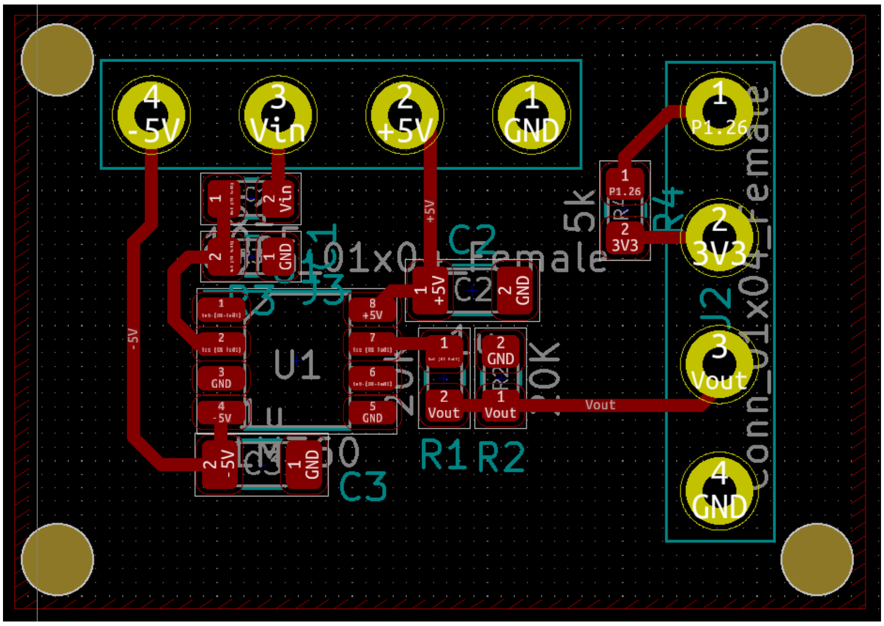

Figure 33: PCB without ground plane.

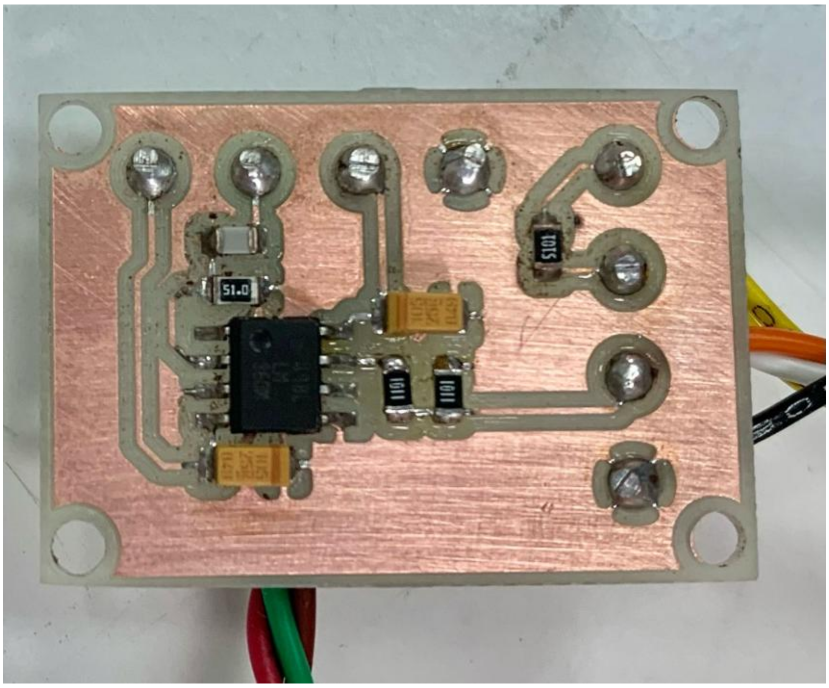

Figure 34.

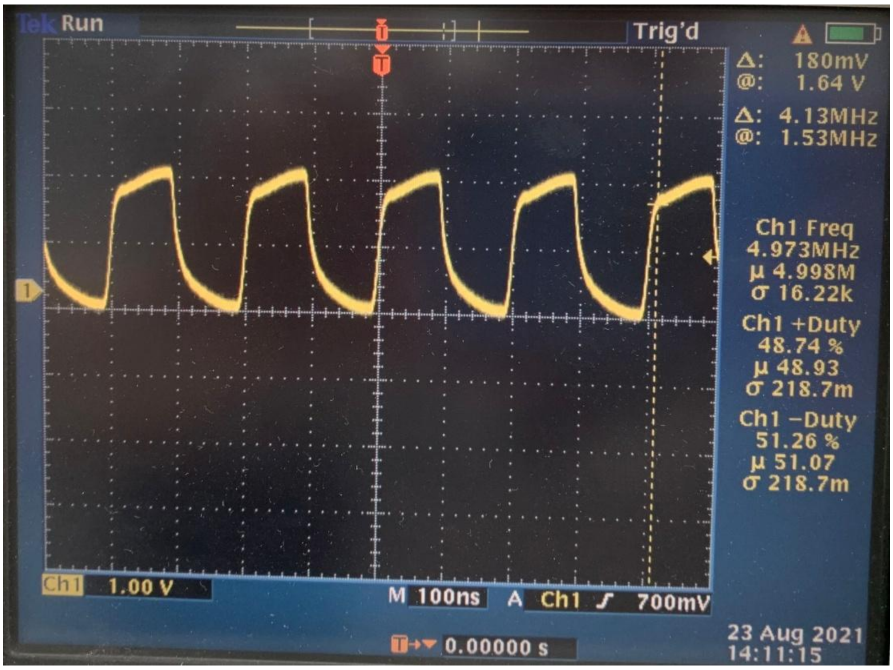

Figure 35: output value of the board, that is a 0 to 2 V 5MHz signal, as is required.