

LABORATÓRIO 1 – MICROCONTROLADORES 2

Revisão Keil – Funções / temporizações

Objetivo:

Desenvolver a função de um controlador de velocidade.

Tarefas:

Criar um novo projeto e implementar um sistema de controle de velocidade que registre quando um limite for ultrapassado. O programa deve ser descarregado e testado passo a passo através do modo debug do kit. A avaliação será realizada ao final da aula do laboratório e considerará a participação do aluno nas atividades do laboratório e a execução das tarefas.

S1----- 10 metros ----- S2

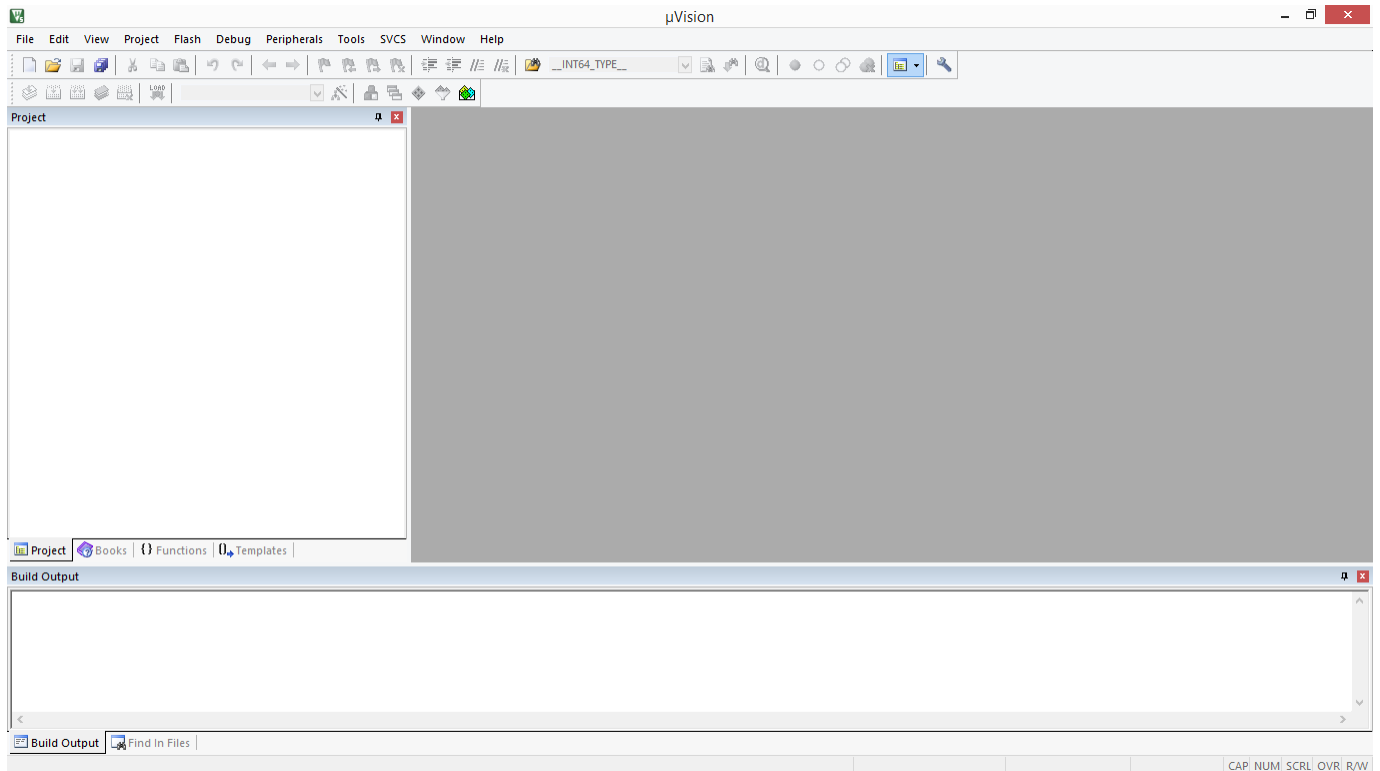
Sinal luminoso – L1

Limite de velocidade: **72Km/h**

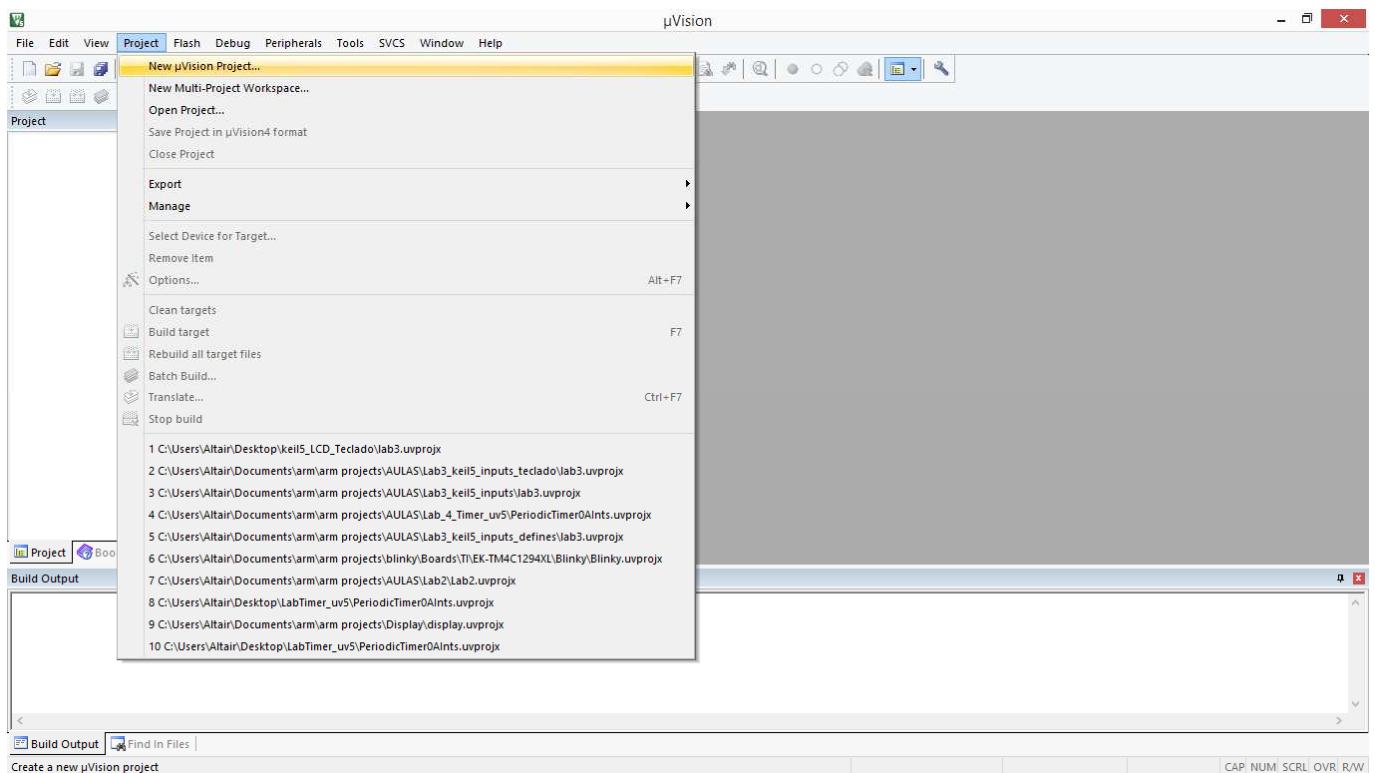
Roteiro para criar e compilar um projeto no KEIL em linguagem C:

Item A: Para criar um novo projeto

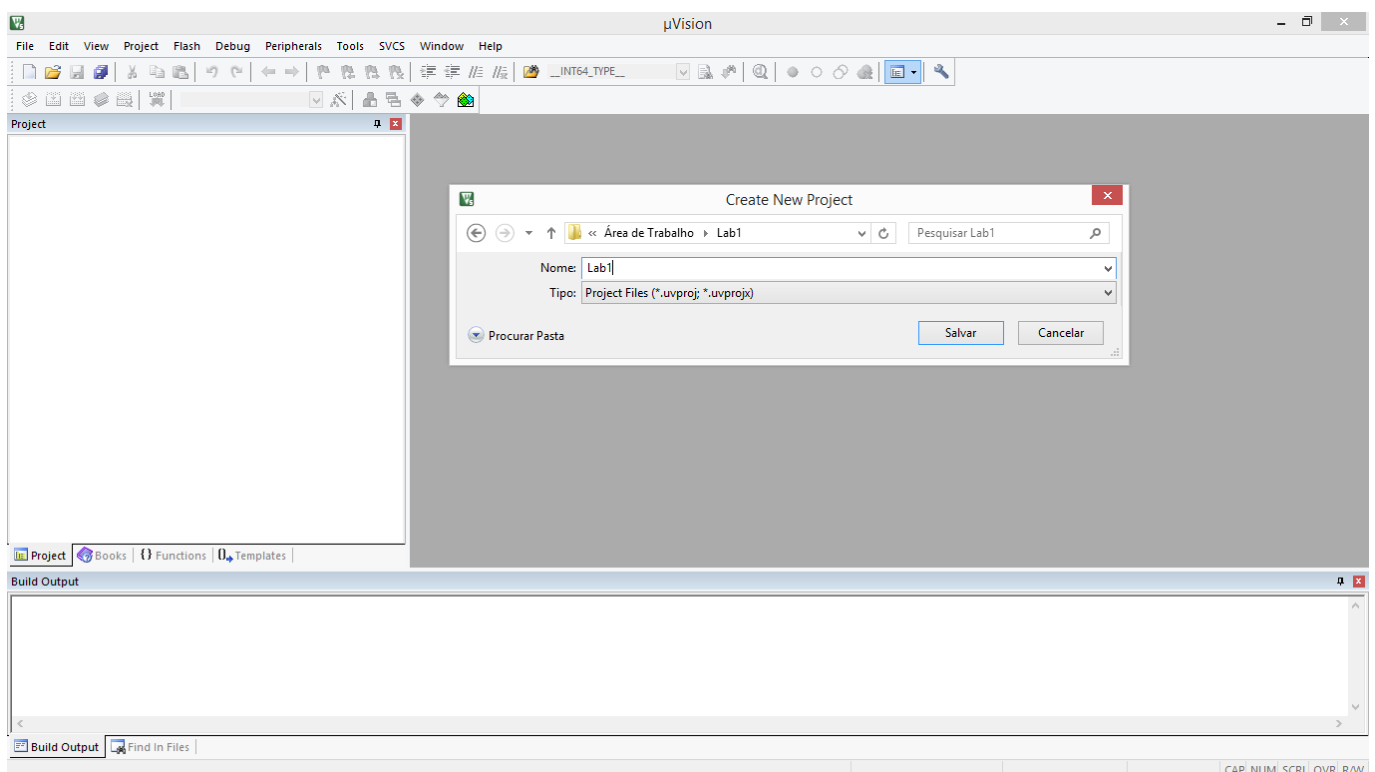
1. Abrir o programa KEIL



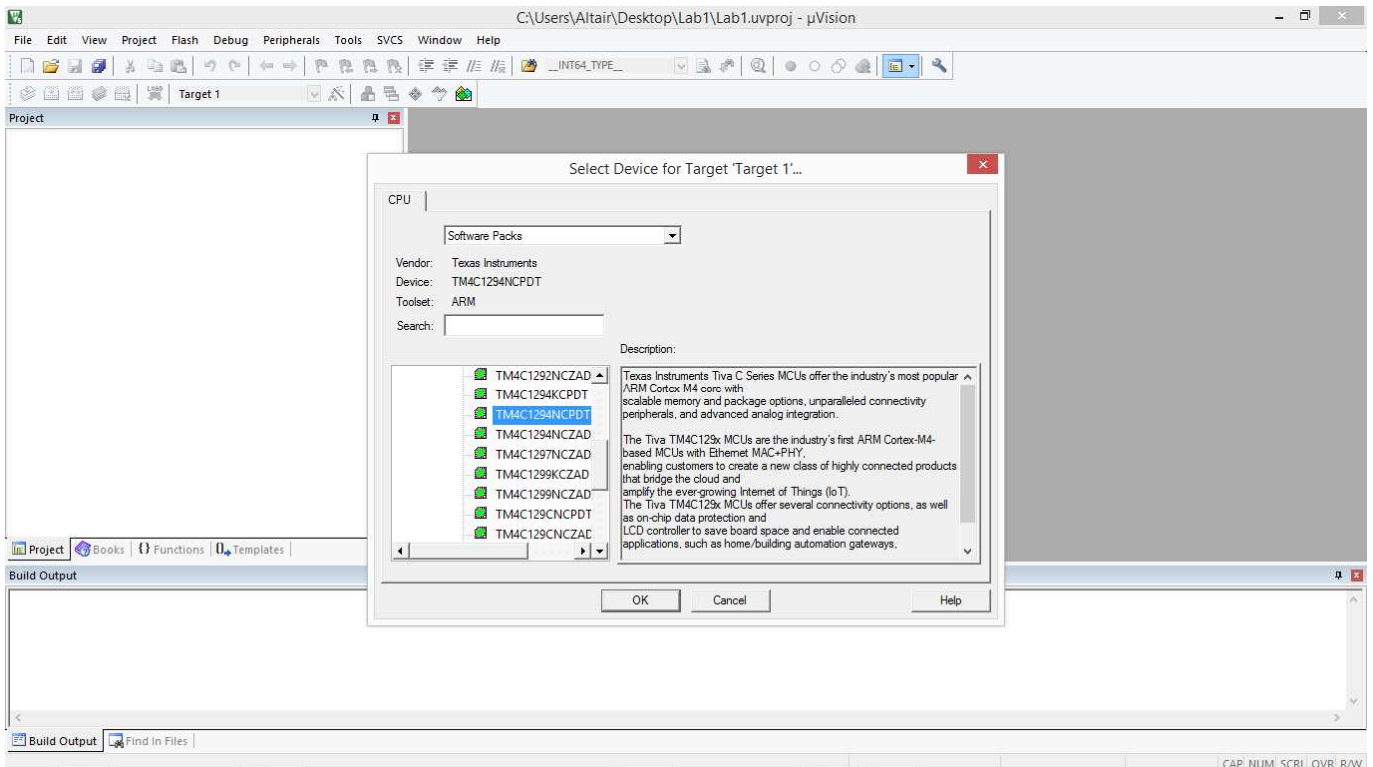
2. Clicar em Project -> New uVision project



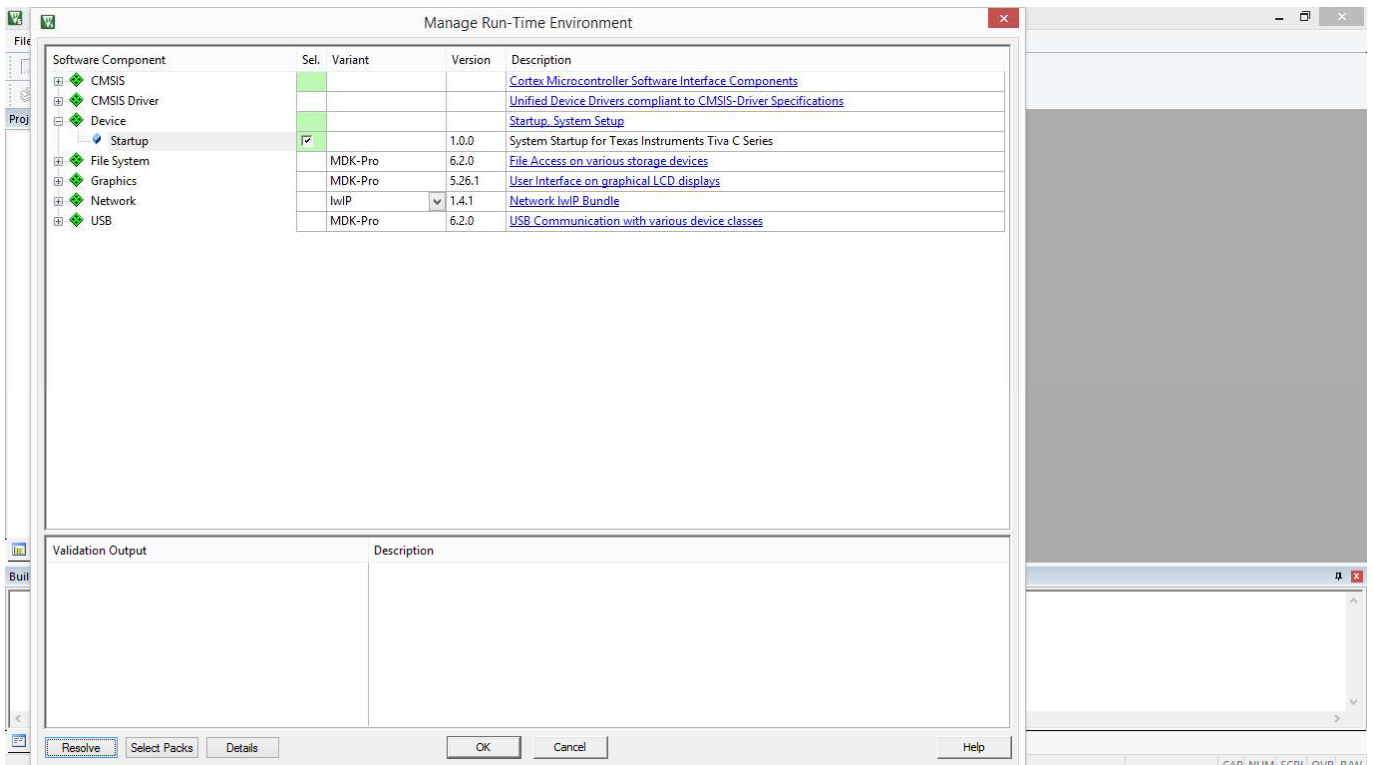
3. Criar uma nova pasta “Lab1” e inserir no nome do Project também “Lab1”:



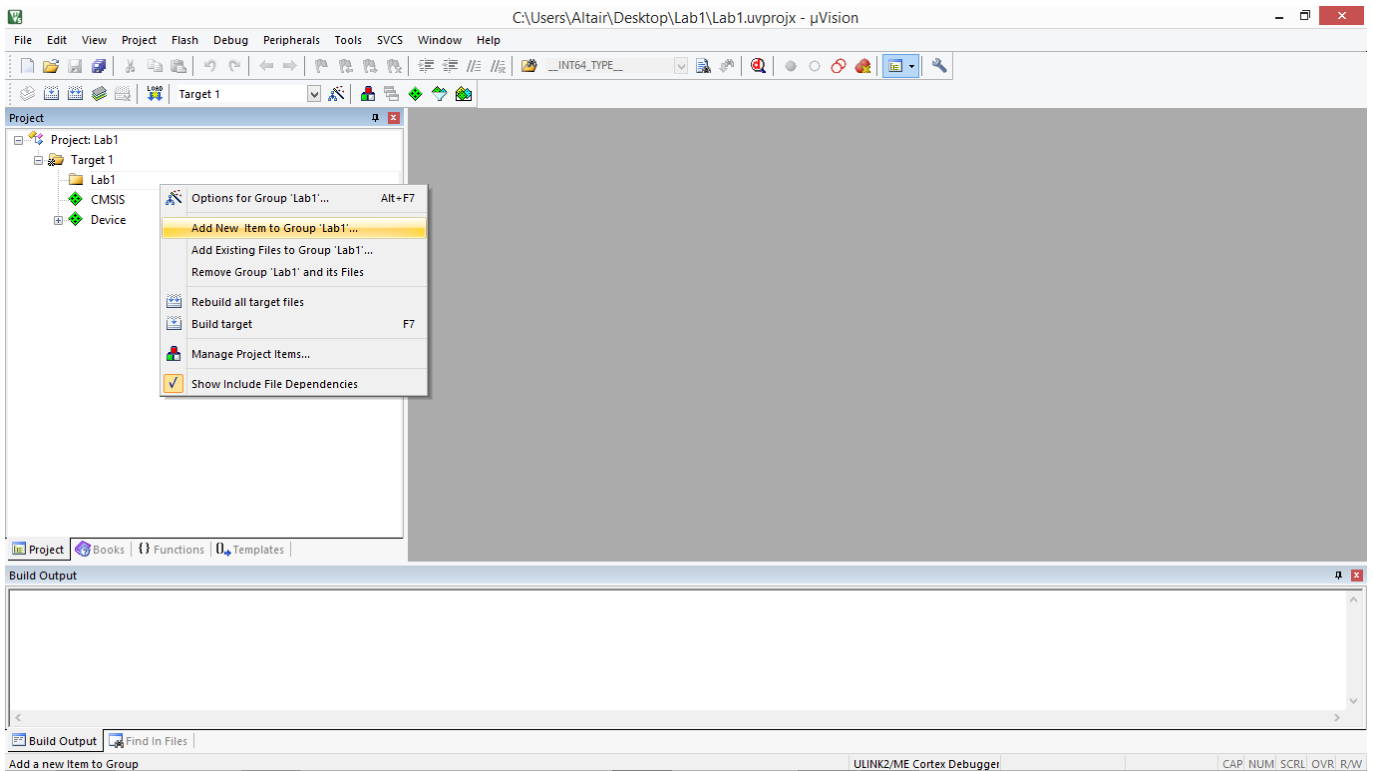
4. Escolher o microcontrolador TM4C1294NCPDT. Caso não o encontre na lista siga as instruções do Item C:



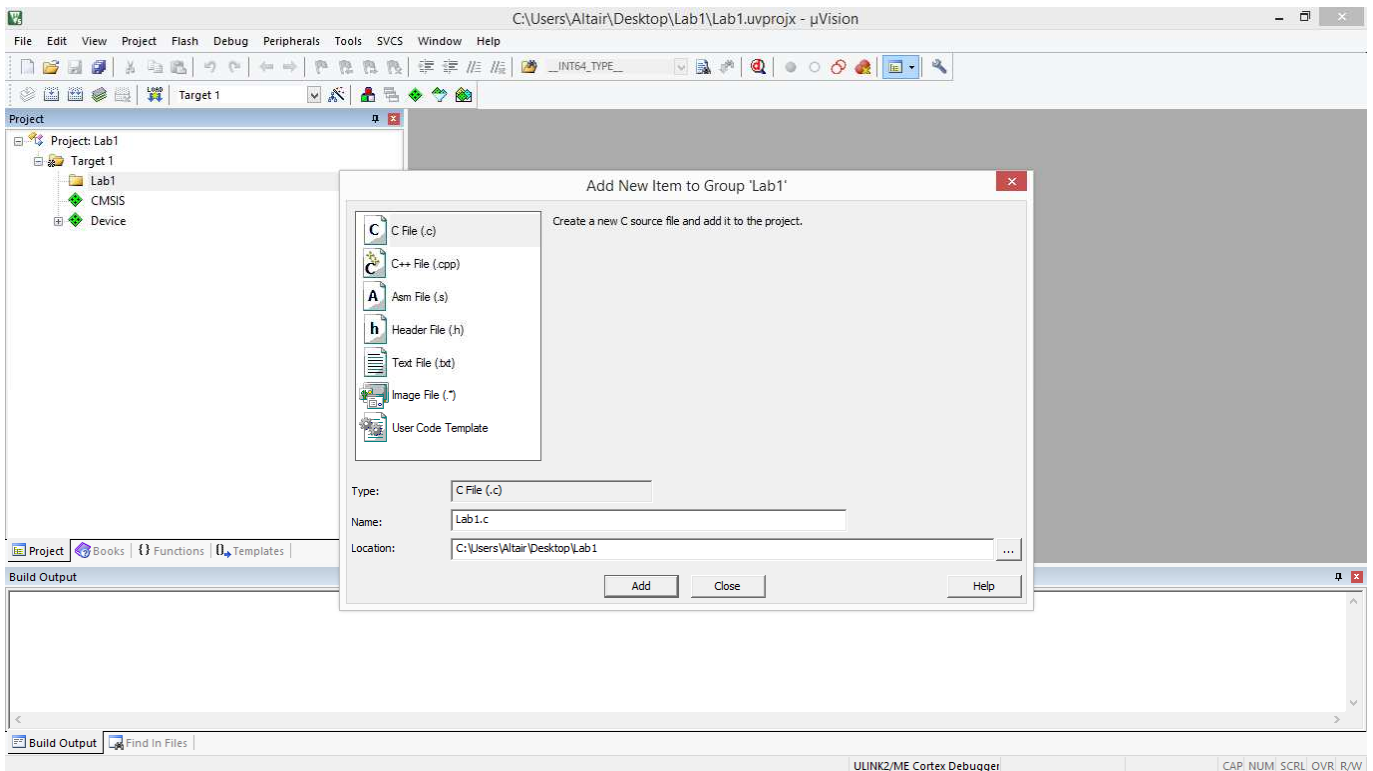
5. Selecione Device->Startup e Clique em "Resolve" e depois "OK":



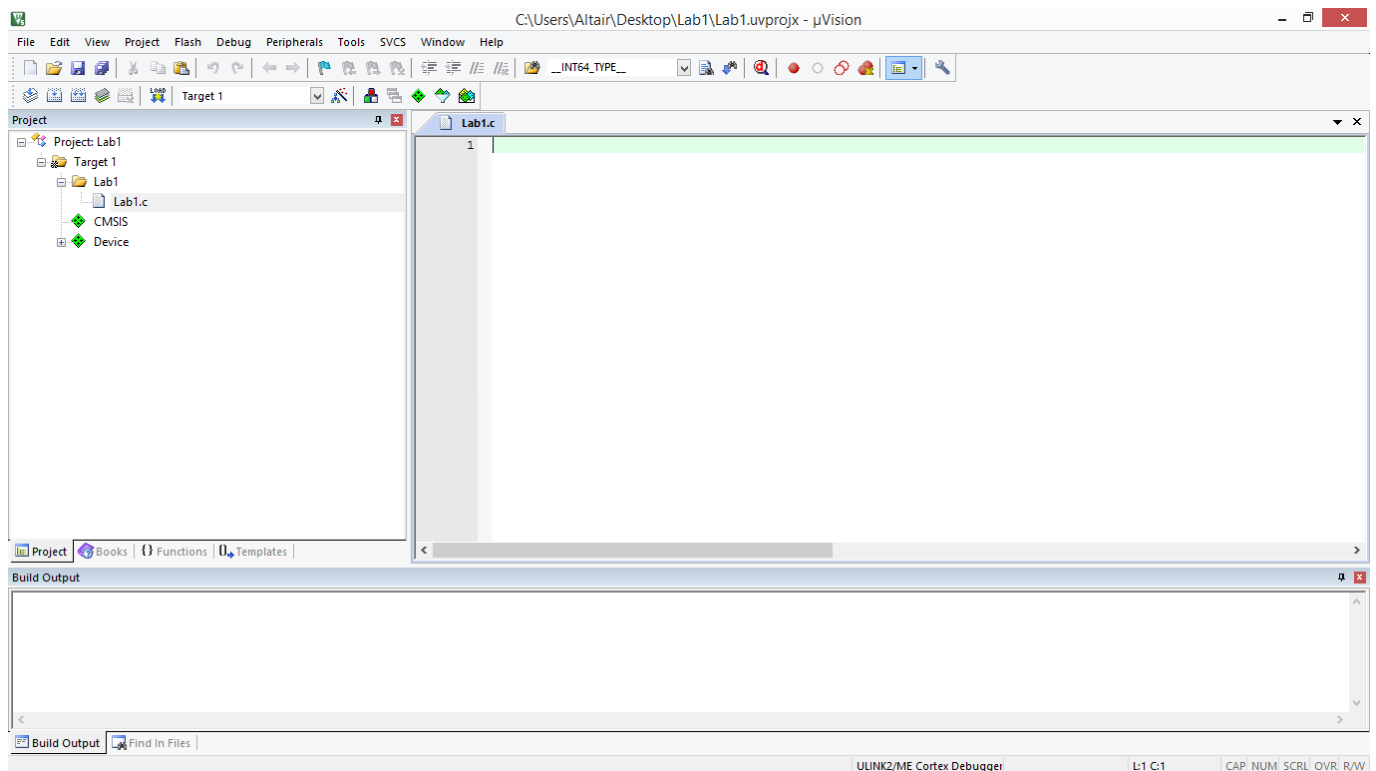
6. Renomear o “New Group 1” para Lab1 e clicar em “Add New Item to Group “Lab1” ...”:



7. Escolha “C File (.c)” e entre com o nome “lab1.c”:

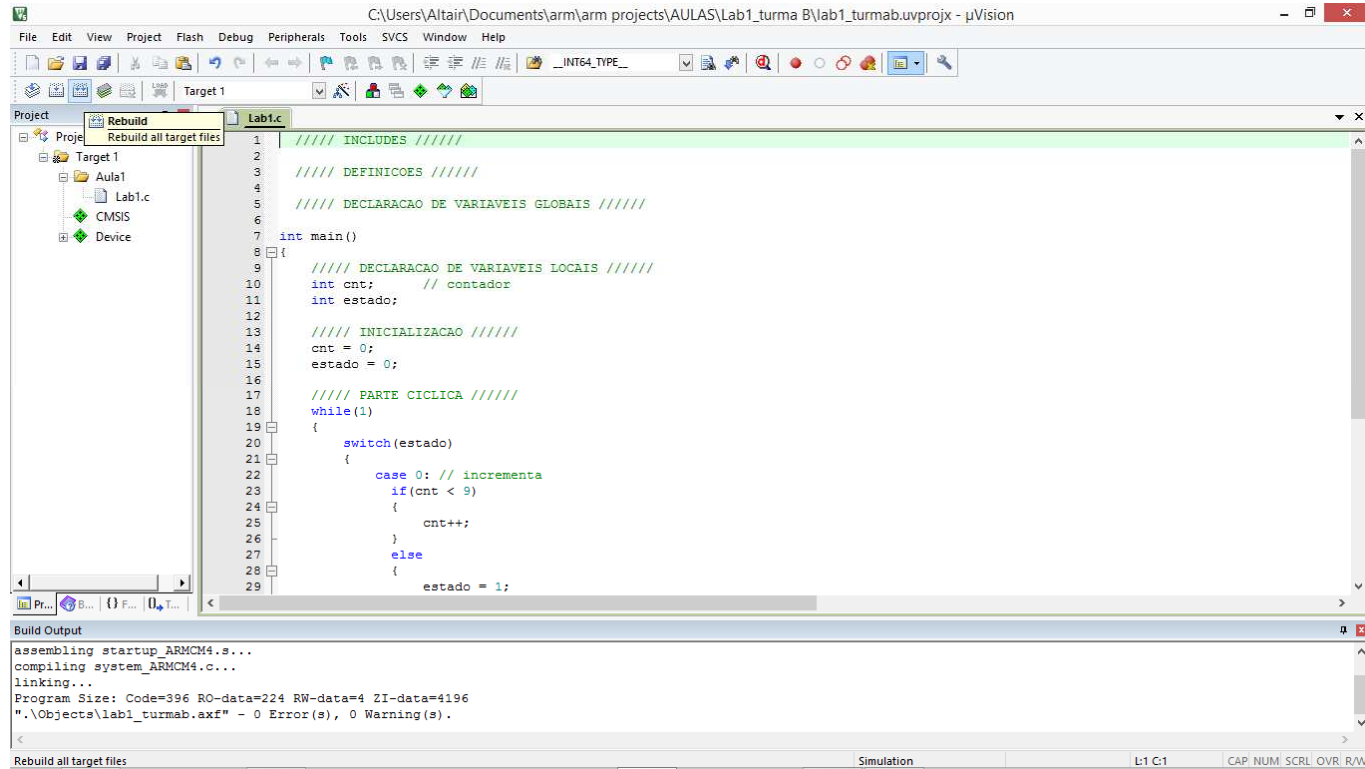


8 . Pronto, você já pode começar a codificar aqui.

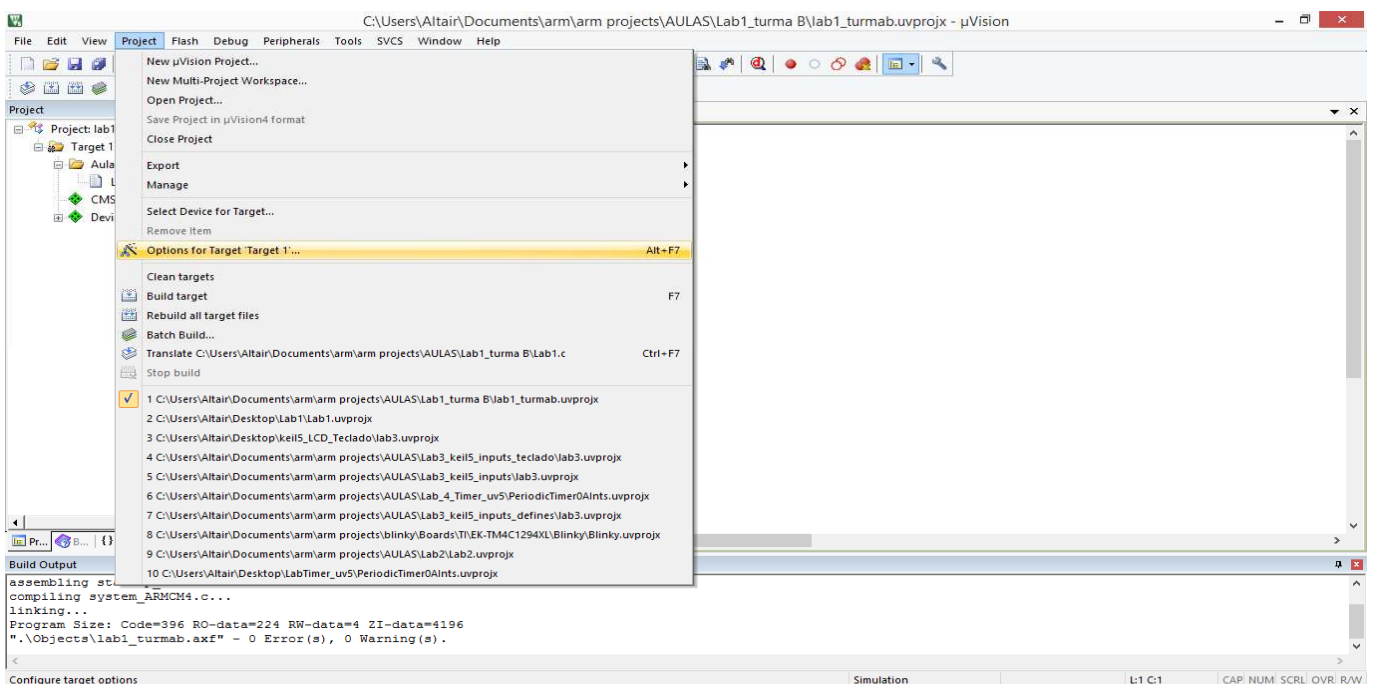


Item B: Para compilar e simular

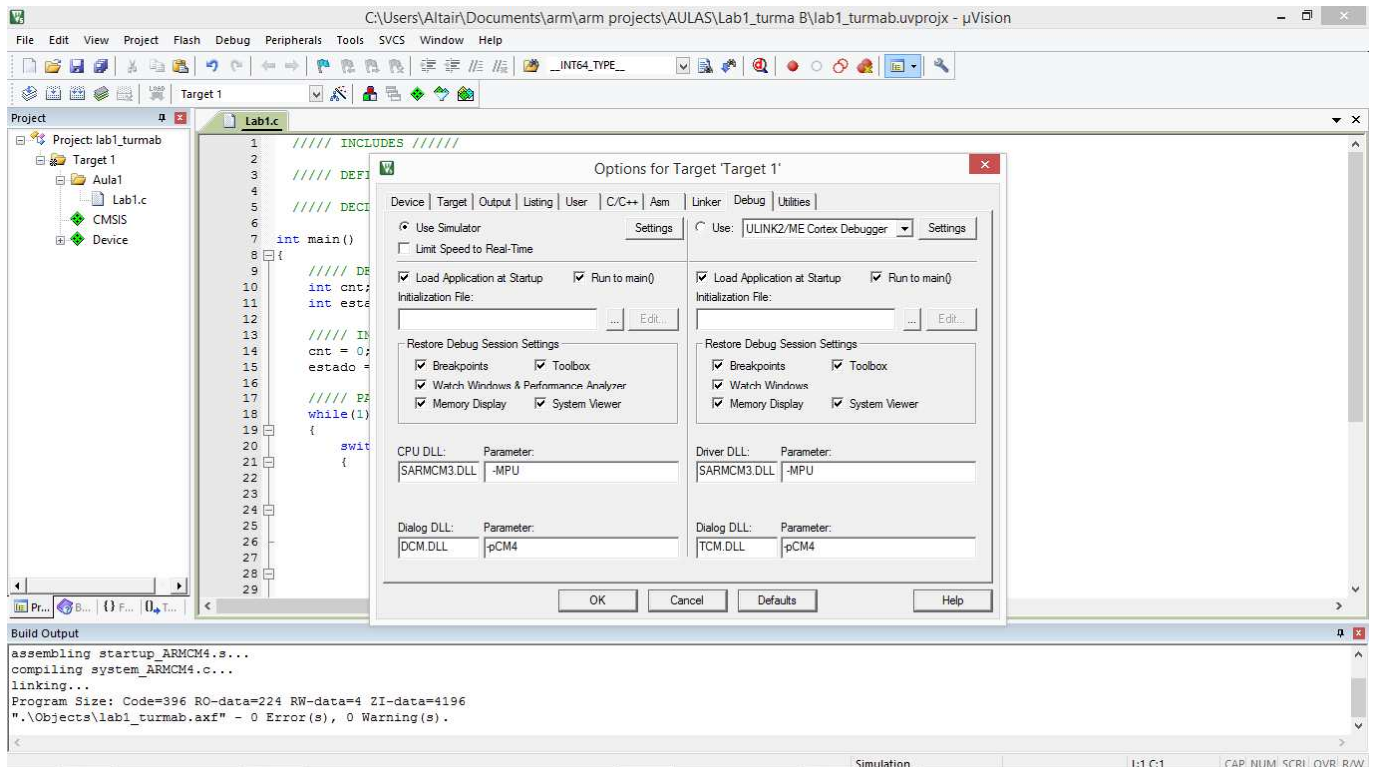
1. Após finalizar o programa, clicar no ícone “Rebuild”. Para que a compilação seja realizada com sucesso deve-se obter 0 Errors(s), 0 Warning(s), como abaixo:



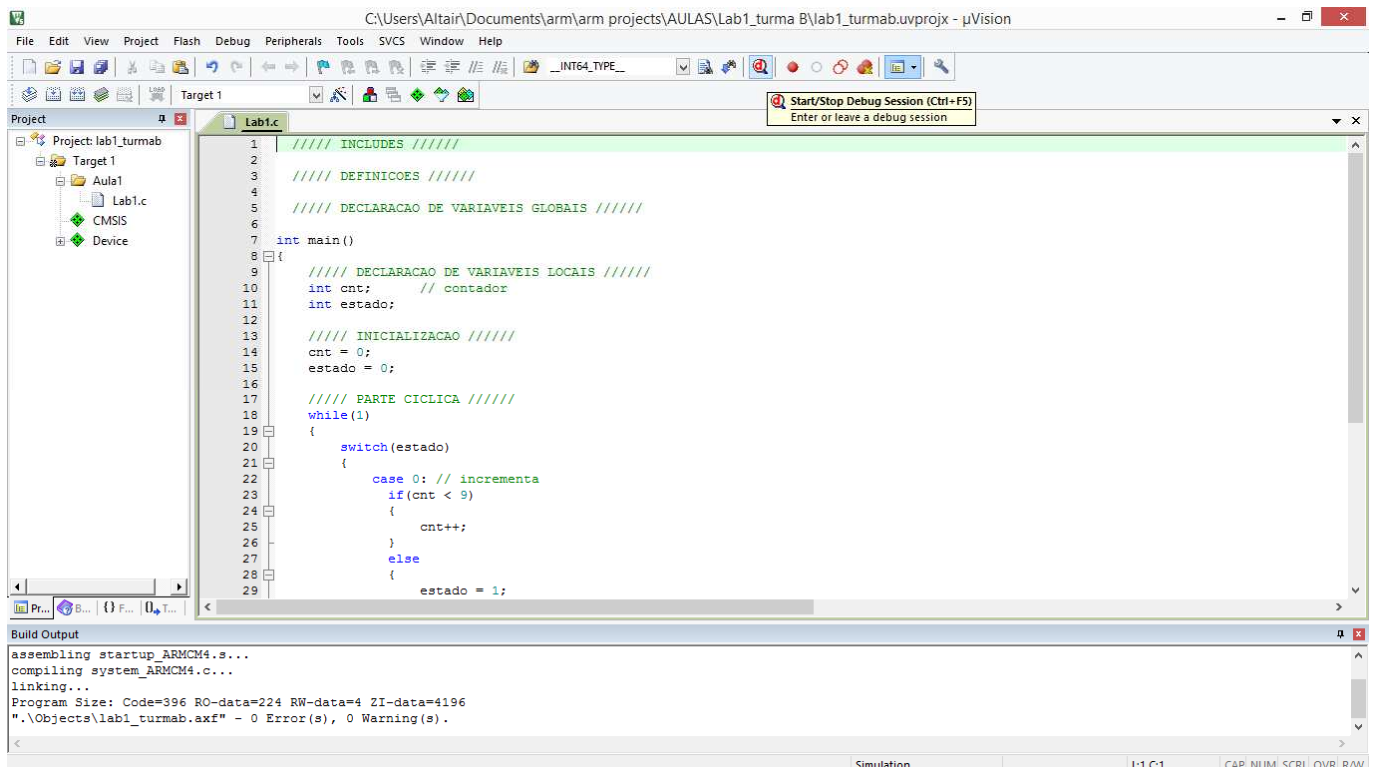
2. Selecionar Project -> Option for Target 'Target 1' :



3. Em DEBUG selecionar “Use Simulator”, para selecionar o modo simulador:



4. Clique no ícone “Start/Stop Debug Session (Ctrl+F5)”:



5. Utilize os ícones abaixo para navegar durante a execução:



6. Insira breakpoints para ativar paradas de execução onde desejar:

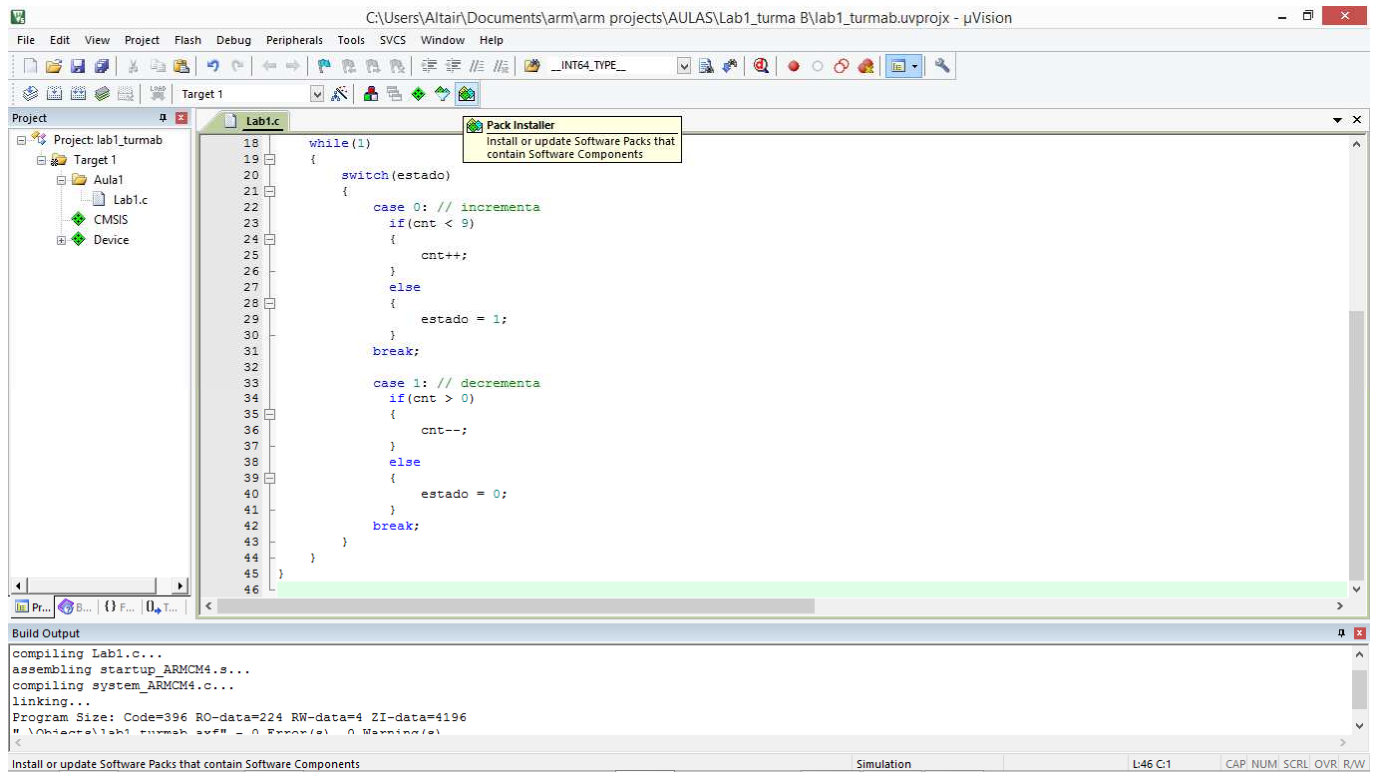


7. Utilize da janela “Watch” para visualizar os valores de variáveis.

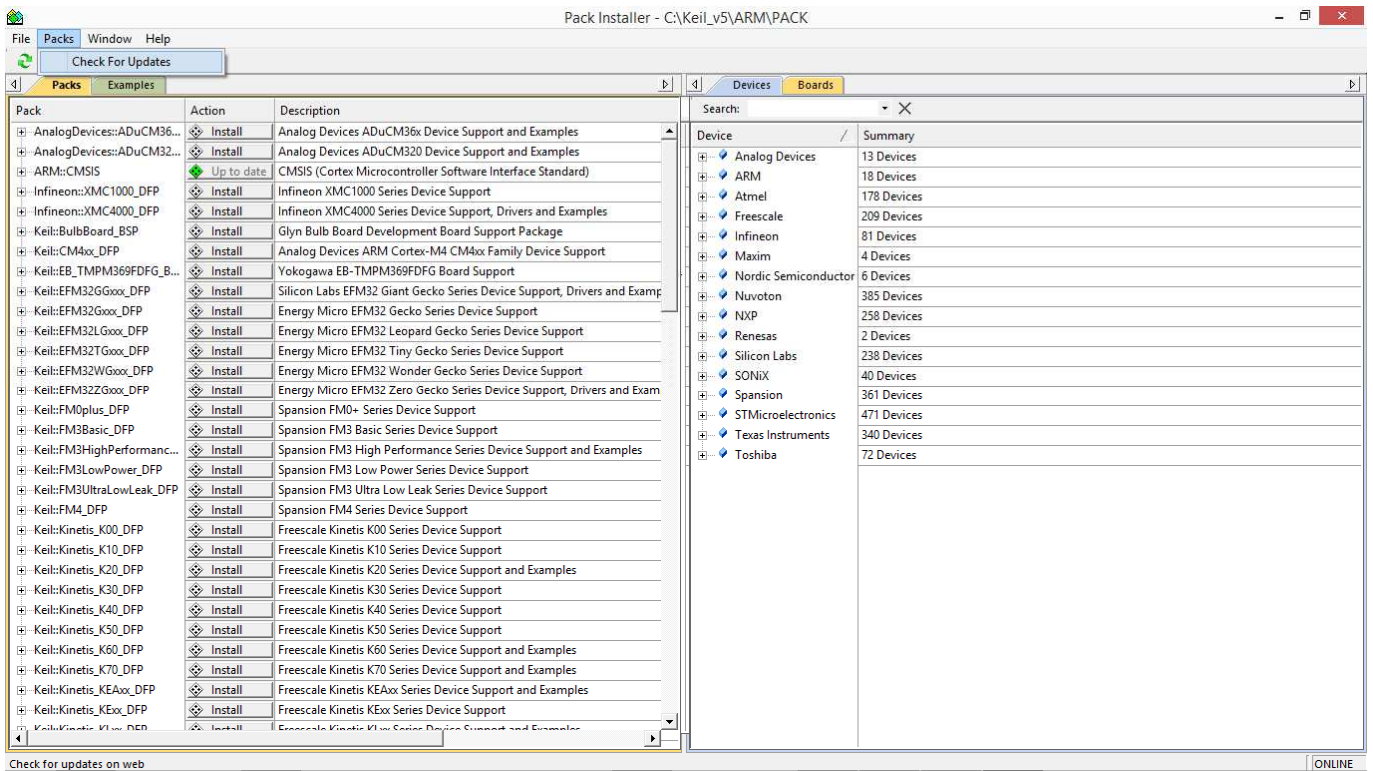
Watch 1		
Name	Value	Type
cnt	0	int
<Enter expression>		

Item C: Instalando pacotes

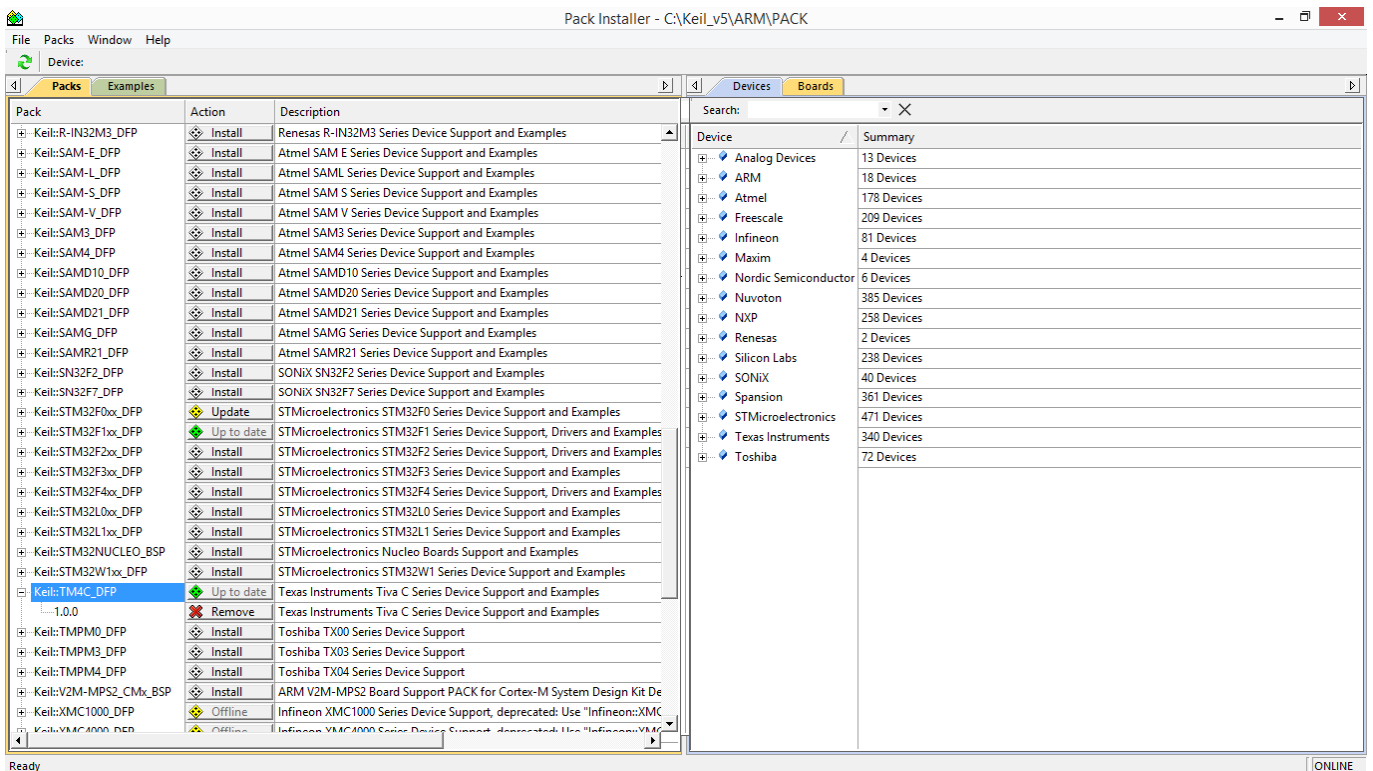
1. Para instalar novos pacotes, clique em “Pack Installer”:



2. Clicar em Packs -> Check for Updates e a instalação de pacotes de será executada automaticamente.



3. Observe que o item Keil: TM4C_DFP agora encontra-se instalado.



- O programa abaixo pode ser usado como referencia. Obs.: este programa está incompleto e possui erros!!!

```
/*-----
PROVA P2
*-----*/
#include <TM4C129.h>

/*-----
Defines
*-----*/
#define PN0          (1UL << 0)

#define PJ0          (1UL << 0)
#define PJ1          (1UL << 1)

#define PortN        GPION->DATA
#define PortF        GPIOF_AHB->DATA
/*-----
Global Variables
*-----*/
long                TimerUpCnt;
char                S1, S2;
char                step;
long                Time;

void SysTick_Handler(void) {
    TimerUpCnt ++;
}

int main()
{
    // Local variables

    /***** INIT *****/
    /* Enable clock and init GPIO outputs */
    SYSCTL->RCGCGPIO |= (1UL << 12);
    GPION->DIR        |= PN0;
    GPION->              |= PN0;

    SYSCTL->RCGCGPIO |= (1ul << 8);
    GPIOJ_AHB->PUR    |= PJ0 | PJ1;
    GPIOJ_AHB->DIR    &= ~(PJ0 | PJ1);
    GPIOJ_AHB->DEN    |= (PJ0 | PJ1);

    SystemCoreClockUpdate();
    SysTick_Config(SystemCoreClock / 1000);
```

```
/****** CYCLIC *****/
while(1)
{
    // read inputs
    S1 = (((GPIOJ_AHB->DATA) & PJ0)) ? 0:1;
    S2 = (((GPIOJ_AHB->DATA) & PJ1)) ? 0:1;

    switch(step);
    {
        case 0:
            if(S1)
            {
                TimerUpCnt = 0;
                step = 1;
            }

        case 1:
            if(S2)
            {
                if(TimerUpCnt <= 500)
                {
                    GPION->DATA |= PN0;
                    TimerUpCnt = 0;
                    step = 2;
                }
            }

            break;
        case 2:
            if(TimerUpCnt >= 1000)
            {
                GPION->DATA &= ~PN0;
                step = 0;
            }

            break;
    } // switch(step)
} // while(1)
}
```