

Ambientes de desenvolvimento para softwares embarcados

Buildroot, OpenEmbedded/Yocto

Phelipe Wesley de Oliveira¹

¹ Universidade Federal do Ceará



Índice

1 Introdução

2 Linux Embarcado

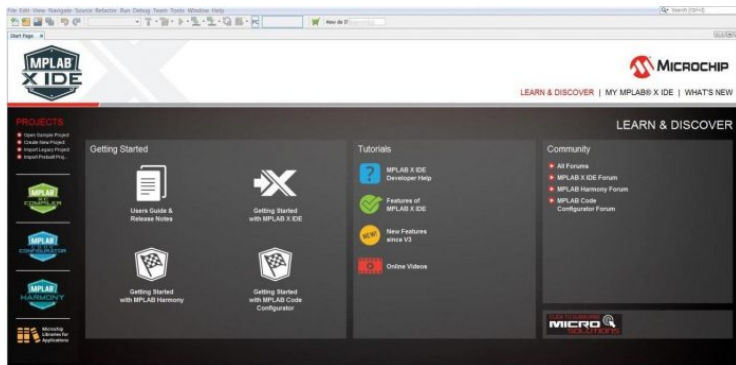
Sistema Embarcado

- Um sistema embarcado, ou sistema embutido, é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla.
- Um sistema embarcado realiza um conjunto de tarefas pré-definidas, geralmente com requisitos específicos.
- Já que o sistema é dedicado à tarefas específicas, pode-se otimizar o sistema reduzindo tamanho, recursos computacionais e custo do produto.

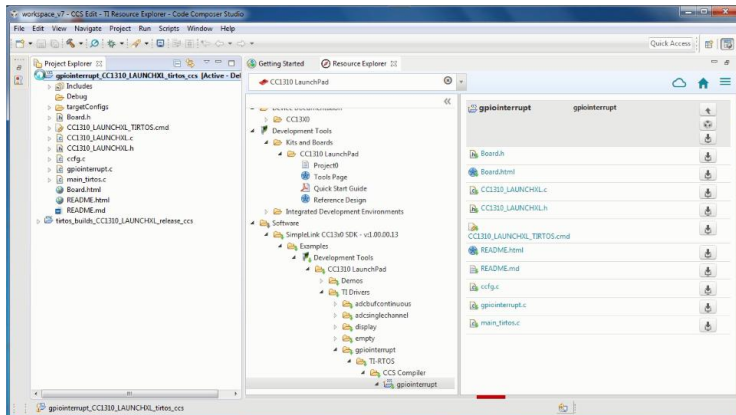
Sistema Embarcado

- Costumam fazer uso de microcontroladores.
- É necessário programar os microcontroladores.
- Existem alguns ambientes de desenvolvimento que facilitam essa tarefa.

MPLAB-X IDE



CCS IDE



Arduino IDE

DHT11_humidity Arduino 1.6.12	DHT11_humidity Arduino 1.0.6
<pre>1 #include <LiquidCrystal.h> 2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); 3 #define DHT11_PIN 0 // ADC0 4 5 byte read_dht11_dat() 6 { 7 byte i = 0; 8 byte result=0; 9 for(i=0; i< 8; i++){ 10 11 12 while(!(PINC & _BV(DHT11_PIN))); // wait for 50us 13 delayMicroseconds(30); 14 15 if(PINC & _BV(DHT11_PIN)) 16 result = (1<<(7-i)); 17 18 }</pre>	<pre>1 #include <LiquidCrystal.h> 2 LiquidCrystal lcd(12, 11, 5, 4, 3, 2); 3 #define DHT11_PIN 0 // ADC0 4 5 byte read_dht11_dat() 6 { 7 byte i = 0; 8 byte result=0; 9 for(i=0; i< 8; i++){ 10 11 12 while(!(PINC & _BV(DHT11_PIN))); // wait for 50us 13 delayMicroseconds(30); 14 15 if(PINC & _BV(DHT11_PIN)) 16 result = (1<<(7-i)); 17 while(!(PINC & _BV(DHT11_PIN))); // wait '1' finish 18 }</pre>
1 NodeMCU 1.0 (ESP-12E Module), 80 Mhz, 256000, 4M (GM SPIFFS) on COM15	1 Arduino Mega (ATmega1280) on COM7

Platformio IDE

The screenshot displays the Platformio IDE interface with the following components:

- Left Sidebar:** Project Explorer showing the file structure of the 'PlatformIO' project, including 'lib', 'readme.txt', 'src', and 'G- blink.cpp'.
- Main Editor:** Displays the 'G- blink.cpp' file. The code includes an Arduino-style setup function that initializes an LED pin and writes a high signal. A tooltip for 'Intelligent Code Completion' is visible over the 'Serial.begin' call.
- Bottom Panel:** Contains three sub-panels:
 - PlatformIO: Debug:** Shows the current debug session status.
 - Smart Code Linter:** Displays a list of linting errors and warnings. One error is shown: 'call_undefined' was not declared in this scope at line 14:18.
 - PIO Debug:** Shows the current debug session status, including the target (sre/blink.cpp), version (15:14), and the current thread (#0 in digitalWrite() at wiring_digital.c:82).
- Right Sidebar:** Contains two panels:
 - Watch Variables:** A table showing the values of variables being watched during the debug session. The table has columns for 'Expression', 'WP', and 'Value'. Variables include 'Serial', 'HardwareSerial', 'sercom', 'I2CM', 'I2CS', 'SPI', 'USART', 'rxBuffer', 'JHead', 'JTail', 'uc_phnRX', 'uc_phnTX', 'uc_padRX', 'SERCOM', 'uc_phnTX', 'LED_BU', and 'SERCOM'.
 - Breakpoints:** A list of breakpoints set in the code, including one at 'loop()' at 'blink.cpp:22'.
 - Call Stacks:** A list of call stacks, including one for 'Thread #1' and another for '#0 in digitalWrite() at wiring_digital.c:82'.

Linux Embarcado

- Existem aplicações onde somente o uso do microprocessador não é suficiente.
- Certas aplicações necessitam de um sistema operacional.
- O uso de um sistema operacional igual ao dos computadores domésticos é inviável pelo excesso de pacotes e funcionalidades desnecessárias.

Linux Embarcado

- É Necessário criar um sistema operacional customizado para o hardware utilizado.
- O Linux é comumente utilizado nessas atividades.

Linux Embarcado