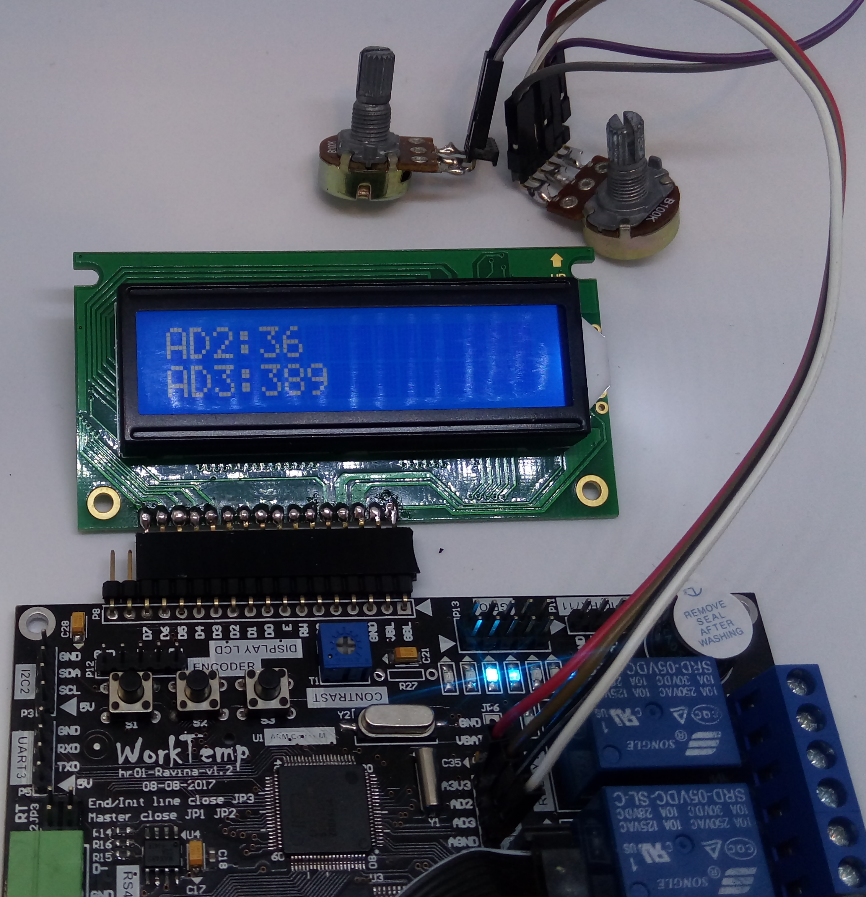
**Utilizando o ADC**

Há inúmeras aplicações do uso do conversor analógico digital (ADC – Analogic Digital Converter). Contudo, o princípio de leitura é o mesmo para maioria das aplicações, e neste exemplo vamos usar um potenciômetro para gerar um sinal analógico e ler os canais analógicos isoladamente aplicando os seus devidos filtros:



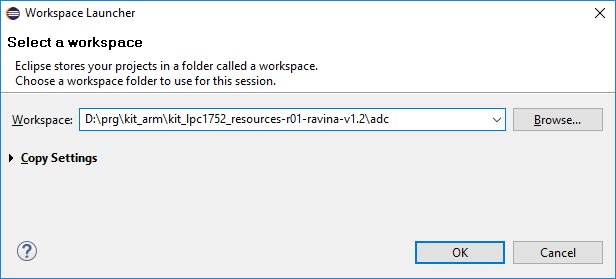
**Conexão com o Kit ARM**

Os sinais analógicos são lidos no conector P9 do kit, e os potenciômetros são conectados nele com a seguinte conexão:

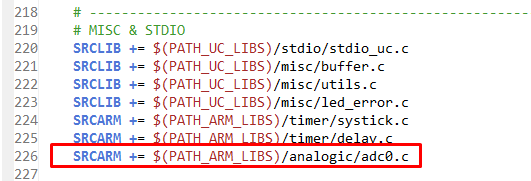
|  |  |
| --- | --- |
| **Potenciômetro 1** | **Potenciômetro 2** |
| Terminal central ao P9 pino 2 | Terminal central ao P9 pino 3 |
| Extremidade 1 ao P9 pino 1 | Extremidade 1 ao P9 pino 1 |
| Extremidade 2 ao P9 pino 4 | Extremidade 2 ao P9 pino 4 |

**Programa Exemplo**

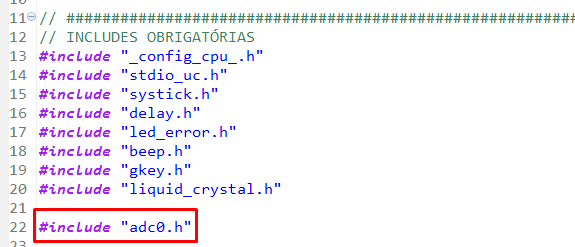
Abra o exemplo da pasta “adc” com o Eclipse:



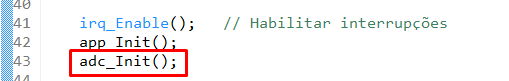
No makefile, na secção de inserção de fontes, foi incluindo o adc0.c :



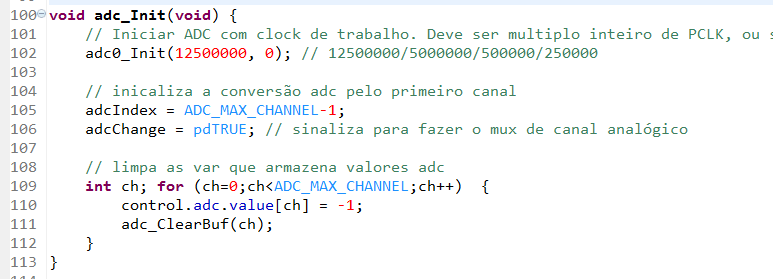
Em app.h foi adicionado a include do adc para usar as suas funções na aplicação:



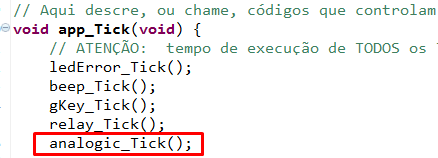
Em main.c dentro do int main(void), é inicializado o gerenciador adc da aplicação:



O gerenciador adc da aplicação inicializa o ADC do ARM com velocidade de conversão de 12.5Mhz, e inicializa os filtros de leitura dos canais analógicos:



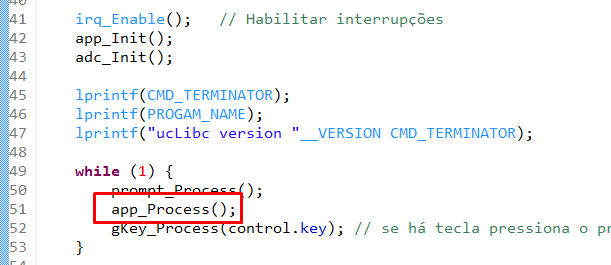
As leituras dos ADCs são feitas em tempos regulares preciso chamar a função analogic\_Tick dentro da função app\_Tick. A função app\_Tick chama todas as funções ticks do sistema:



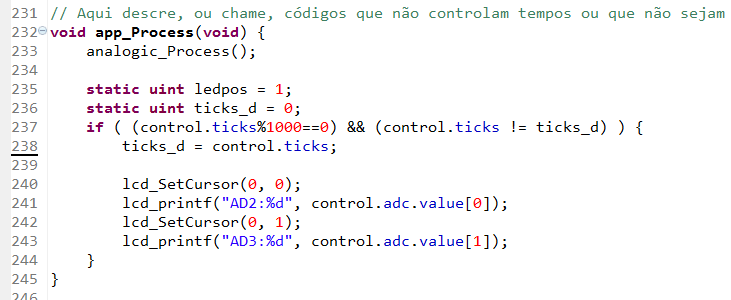
A função app\_Tick é chamada a cada 1ms, isto foi especificado no início do programa na função main:



O processo de leitura dos canais analógicos é feito pelo app\_Process dentro do loop da função main:



Em app\_Process chama o processo analógico que faz as leituras dos canais analógicos e mostra os valores lidos no display LCD:



Em analogic\_Process faz a leitura dos canais analógicos e aplica os devidos filtros. O filtro basicamente faz as médias das leituras e descarta os valores fora dos padrões lidos.

**Compile e Execute**

Com tudo instalado e configurado, compile a aplicação, grave e execute no ARM. Use o CLI junto ao um terminal para enviar comandos ao ARM. Abaixo mostra a leitura dos canais quando emitido o comando de status:

