

Esse artigo foi escrito pelo engenheiro Ismael Lopes da Silva, exclusivamente para o site "www.embarcados.com.br". O link para o artigo é "<https://www.embarcados.com.br/blue-pill-stm32f103c8t6-sofware-bibliotecas-e-ferramentas/>".

No terceiro artigo da série "A Blue Pill", daremos continuidade no estudo. A dica é acompanhar a série desde o primeiro artigo porque é um caminho estruturado, onde a sequência traz benefícios no aprendizado (linha de raciocínio). Nesse artigo veremos o software, bibliotecas, requisitos e ferramentas que utilizaremos nesse aprendizado.

FreeRTOS

Ao contrário da popular família de chips AVR, agora de propriedade da Microchip, a família STM32F103 possui SRAM, RAM estática, suficiente para executar confortavelmente o FreeRTOS (www.freertos.org). O acesso a um Sistema Operacional em Tempo Real (RTOS) oferece várias vantagens, incluindo:

- ✓ Tarefas múltiplas e agendamento;
- ✓ Fila de mensagens;
- ✓ Semáforos e mutexes;
- ✓ Temporizadores.

Uma vantagem particular é a capacidade de multitarefa. Isso facilita consideravelmente o projeto do software. Muitos projetos avançados do Arduino são sobrecarregados pelo uso do estado de máquinas no modelo de loop de eventos. Cada vez que passa pelo loop, o software deve pesquisar se um evento ocorreu e determinar se é hora de alguma ação. Isto exige gerenciamento de estado das variáveis, que rapidamente se torna complexo e leva a erros de programação. Por outro lado, a multitarefa preemptiva fornece controle de tarefas separadas que claramente implementa suas funções independentes. Esta é uma forma comprovada de abstração de software.

Preemptividade, algumas vezes preempção, é o ato de interromper temporariamente uma tarefa sendo executada por um sistema operacional, sem exigir sua cooperação, e com a intenção de retomar à tarefa posteriormente.

O FreeRTOS fornece multitarefa preemptiva, que automaticamente compartilha o tempo da CPU entre as tarefas configuradas. Tarefas independentes, no entanto, acrescentam alguma responsabilidade pela interação segura entre elas. É por isso que o FreeRTOS também fornece filas de mensagens, semáforos, mutexes e muito mais para gerenciar isso com segurança. Vamos explorar os recursos do RTOS ao longo dessa série.

Libopencm3

O desenvolvimento de código para aplicativos MCU pode demandar muita programação. Uma parte desse desafio está se desenvolvendo com o "bare metal" da plataforma. Isso inclui todos os registradores dos periféricos especializados e seus endereços. Além disso, muitos periféricos exigem um certo "rebolado" para torná-los prontos para uso.

É aqui que a biblioteca libopencm3 se encaixa. Não apenas define os endereços de memória dos registradores dos periféricos, mas, também define macros para constantes especiais que são necessárias. Por fim, a biblioteca inclui funções C testadas para interagir com os recursos dos

periféricos de hardware. O uso da biblioteca libopencm3 nos poupa de ter que fazer tudo isso do zero.

Sem Ambiente de Interface de Desenvolvimento

Como já comentado essa série desenvolve projetos com base nas seguintes ferramentas e bibliotecas de código aberto:

- ✓ gcc / g ++ (compiladores GNU: código aberto);
- ✓ Make (GNU binutils: código aberto);
- ✓ Libopencm3 (biblioteca: código aberto);
- ✓ FreeRTOS (biblioteca: código aberto e gratuito para uso comercial).

Com essa base, os projetos devem permanecer utilizáveis por muito tempo. Além disso, permite que usuários do Linux e dos usuários da plataforma Windows, usem o conteúdo desse documento. Se você usa o Windows, pode fazer o download e instalar o ambiente Cygwin, no site www.cygwin.com, porque é simulado um ambiente semelhante ao Linux para a construção dos projetos.

Todos os projetos dessa série fazem uso do utilitário de criação GNU, que fornece várias funções de construção com o mínimo esforço. Se as compilações fornecidas apresentarem erros, certifique-se de usar o comando GNU make.

Framework (Estrutura) de Desenvolvimento

Embora seja possível fazer com que o gcc, o libopencm3 e o FreeRTOS funcionem juntos, isso exige uma quantidade razoável de organização e esforço. Quanto vale o seu tempo? Em vez de fazer esse trabalho tedioso, um framework de desenvolvimento está disponível gratuitamente no github.com para download. Essa estrutura integra o libopencm3 ao FreeRTOS para você.

Também são fornecidos os arquivos make necessários para construir toda a árvore do projeto de uma só vez ou cada projeto individualmente. Finalmente, existem algumas rotinas de biblioteca de código aberto incluídas isso pode reduzir o tempo de desenvolvimento de seus novos aplicativos. Essa estrutura está incluída no download do github.com.

Suposições sobre você

Este trabalho é direcionado a um público que deseja ir além da experiência do Arduino. Isso se aplica a entusiastas, fabricantes e engenheiros. O software de desenvolvimento usa a linguagem de programação C, portanto, a fluência será útil. Da mesma forma, é assumido algum conhecimento básico em eletrônica digital, no que se refere às interfaces periféricas fornecidas pela plataforma. Teoria adicional pode ser encontrada em áreas como o barramento CAN, por exemplo.

A plataforma STM32 pode ser um desafio para configurar e operar corretamente. Grande parte desse desafio é por causa da extrema configurabilidade dos dispositivos periféricos. Cada parte depende de um clock, que deve ser ativado e um divisor também deve ser configurado. Alguns dispositivos são ainda mais afetados pelas configurações de clock. Por fim, cada periférico em si deve estar ativado e configurado para uso. Você não precisa ser um especialista, porque esses procedimentos serão detalhados e explicados nessa série.

Hobistas e desenvolvedores não precisam se preocupar com as dificuldades. Mesmo quando desafiados, eles devem ser capazes de criar e executar cada um dos experimentos do projeto. À medida que o conhecimento e a confiança aumentam, cada um pode avançar nos tópicos abordados. O framework fornecido também permitirá que você crie novos projetos prontos para uso com o mínimo de esforço.

O que você precisa?

Abordaremos brevemente alguns itens que você pode querer adquirir. Certamente, o número um na lista é a placa de desenvolvimento, Blue Pill. Eu recomendo que você compre Blue Pill que incluam os terminais soldados na PCB, para que você possa usar facilmente num protoboard. Para encontrar ofertas da Blue Pill na internet, basta usar o código de referência STM32F103C8T6 na sua pesquisa. Alguns projetos usam mais do que uma Blue Pill se comunicando através de um barramento CAN.



Figura 1 – PCB Blue Pill

Unidade de programação ST-Link V2

Outro item de hardware essencial é um adaptador de programação. Simplesmente procure na Internet por “ST-Link”. Certifique-se de adquirir o programador “V2”, pois não faz sentido usar uma versão antiga. Compre um ST-Link V2 que inclua os cabos, a menos que você já tenha os cabos.



Figura 2 – Programador DT-Link V2

O dispositivo STM32F103C8T6 pode ser programado de várias maneiras, mas, nessa série usaremos apenas com o programador USB ST-Link V2. Isso simplificará as coisas para você durante o desenvolvimento do projeto e permitirá a depuração remota.

Protoboard

Também será necessário uma placa de ensaio para criar protótipos. O Protoboard é uma maneira sem solda de conectar rapidamente componentes e dispositivos, fazer experimentos e, em seguida, no final do experimento, puxar os fios e desconectá-los do Protoboard. Os projetos dessa série são pequenos, exigindo espaço para um Blue Pill e talvez alguns LEDs e outros componentes pequenos. No entanto, outras poucas experiências, usam três unidades se comunicando através de um barramento CAN.

Eu recomendo que você obtenha um Protoboard grande, isso deixa um pouco de espaço para conexão extra. Como alternativa, você pode simplesmente comprar alguns Protoboards pequenos, embora isso seja menos conveniente.

Fios e Jumpers

Você pode não pensar muito na fiação de um experimento, mas, descobrirá que os fios de Jumper podem fazer uma enorme diferença. Você pode cortar e descascar seus próprios fios de bitola AWG22 ou AWG24, mas, isso pode ser inconveniente e demorado. Seria bem melhor você ter uma pequena caixa de fios de Jumper pronta para usar. Eu recomendo que você obtenha os conjuntos variados para obter cores e comprimentos diferentes.

Capacitores de filtro

Utilize capacitores de qualidade, de 0,1 (uF), como os de filme de poliéster metalizado. Sua tensão de trabalho pode ser tão baixa quanto 16 Volts. Eles devem ser conectados aos terminais de alimentação do Protoboard, entre as trilhas positivo e negativo, para filtrar quaisquer transientes e ruídos de tensão.

Adaptador USB TTL Serial

Este dispositivo será essencial para alguns projetos. Este adaptador serial é usado para comunicar com seu desktop/notebook. Isso permite que você se comunique através de um link serial virtual, via USB.

No mercado existem vários tipos disponíveis, mas, preste atenção para obter um com sinais de controle de fluxo de hardware. Os adaptadores mais baratos não terão esses sinais adicionais (RTS e CTS). Sem os sinais de controle de fluxo de hardware, você não poderá se comunicar em alta velocidade, como 115200 baud, sem perder dados.

Se você estiver executando no Windows, também tenha cuidado ao comprar falsificações de Chip FTDI. Tem relatos de drivers de software da FTDI que travam a comunicação. Sua escolha não precisa incluir o FTDI, mas, se o dispositivo alegar compatibilidade com o FTDI, esteja ciente e verifique o suporte do driver.

Provavelmente você precisará de uma etiqueta presa à extremidade do cabo. Nessa etiqueta conterà os dados necessários para identificar os terminais que são conectados na Blue Pill. Normalmente, USB são dispositivos de 5 Volts e, portanto, são compatíveis com TTL. Observe, no entanto, que um dos recursos dos dispositivos STM32F103 é que muitos dos pinos GPIO são

tolerantes a 5 Volts, mesmo que o MCU opere a partir de uma fonte de + 3,3 Volts. Isso permite o uso desses adaptadores TTL sem causar danos. Mas, veremos isso mais tarde.

Fonte de alimentação

A maioria dos projetos funcionam muito bem com a saída de energia do adaptador USB ou TTL. Mas se o projeto consome mais do que a quantidade usual de corrente, pode ser necessário uma fonte de alimentação. Existem bons adaptadores para serem usados diretamente no Protoboard. Alguns são anunciados na Internet como "Módulo de fonte de alimentação para Protoboard, MB102 sem solda, 3.3V e 5V".



Figura 3 – Módulo de fonte para Protoboard MB102

O módulo MB102 é conveniente porque pode fornecer 3,3 ou 5 Volts. Além disso, inclui um botão liga/desliga. O módulo MB102 aceita até 12 Volts de entrada. A outra consideração está relacionada aos adaptadores que são alimentados pelas tomadas residências e fornecem tensões de, por exemplo, 9V. A saída desses adaptadores alimentam os módulos MB102. Muita atenção com a qualidade desses adaptadores, porque quando estão sem carga sua tensão de saída podem ser suficiente alta para danificar o módulo MB102 e até sua Blue Pill.

Miscelâneos

Existem alguns itens pequenos que você já pode ter, caso contrário, você precisará obter alguns LEDs, resistor SIP-9 e resistores para uso geral. Normalmente, um LED consome cerca de 10 (mA) de corrente para um brilho normal. LEDs menores requerem apenas 2 a 5 (mA). Com uma tensão de alimentação próxima a 3,3 Volts, você precisará de um resistor de cerca de 220 (Ω) para limitar essa corrente. Portanto, pegue alguns resistores de 220 (Ω), 1/8 de watt será suficiente.

Outra componente que você pode considerar é o resistor SIP-9. Se, por exemplo, você quiser acender oito LEDs, precisará de oito resistores limitadores de corrente. Resistores individuais funcionam, mas requerem fiação extra e ocupam espaço na placa de ensaio. O resistor SIP-9, por outro lado, possui uma conexão comum aos oito resistores. As outras oito conexões são a outra extremidade dos resistores internos. Usando esse tipo de componente, você pode otimizar a montagem do projeto.

Conforme a proposta inicial vamos trabalhando nas bases para fundamentar o aprendizado, portanto, aqui concluo o terceiro artigo da série.