

Microcontroladores embarcados

Nesta leitura exploraremos a diversidade de microcontroladores embarcados.

Os microcontroladores ficam na parte inferior da hierarquia de hardware de computação, pelo menos em termos de capacidade computacional. Você deve estar se perguntando o que diferencia um MCU da CPU da máquina à sua frente. Bem, em alguns aspectos um MCU e uma CPU são bastante semelhantes, nomeadamente porque um MCU é a integração de uma CPU de baixo desempenho e outros periféricos num único chip. Desta forma, os MCUs também são semelhantes, embora menos sofisticados, a um System-on-a-Chip (SoC). Os periféricos típicos integrados em um MCU incluem memória, conversores analógico-digital (ADC), conversores digital-analógico (DAC), temporizadores, contadores, IO de uso geral (GPIO), modulação por largura de pulso (PWM), direto acesso à memória (DMA), gerenciadores de interrupção e controladores de protocolo serial. Podemos analisar essa integração e tirar duas conclusões: os MCUs enfatizam as conexões com os ambientes em que vivem e exigem que todo o hardware integrado seja o mais compactado possível. Assim como uma CPU instalada em uma placa-mãe, os MCUs são chips com pacotes soldados a PCBs muito menores que, no mínimo, devem ter circuitos de alimentação de suporte e uma interface de programação. O último pré-requisito pode ser evitado se o MCU for programado antes da montagem, mas isso é incomum, pelo menos no desenvolvimento. Além disso, as especificidades de um sistema embarcado personalizado dependem das necessidades da aplicação. Ao aproveitar um conselho de desenvolvimento, você desejará centralizar sua seleção em atender, se não exceder, os mesmos requisitos. Orientaremos você em uma comparação de placas de desenvolvimento e especificações de MCU em um momento.

Voltando à hierarquia de hardware de computação: os MCUs são claramente superados pelo desempenho do seu computador pessoal médio, bem como pelos clusters de computação. Eles são um tanto adjacentes, se não indiscutivelmente inferiores, ao que chamaremos de computação “intermediária” na forma de computadores de placa única (SBCs), como o Raspberry Pi. SBCs e placas de desenvolvimento MCU são bastante semelhantes conceitualmente e em termos de poder de computação, mas são diferenciados porque os SBCs pretendem ser computadores de placa única de baixo consumo de energia com sistemas operacionais adequados e IO típicos de computadores pessoais, mas muitas vezes não possuem os periféricos que necessitamos. listados acima, tornando-os inadequados no contexto do projeto de sistemas embarcados. Olhando para a hierarquia de computação que apresentamos (clusters > computadores > computação intermediária > microcontroladores), você pode se perguntar por que tanta ênfase é colocada no desenvolvimento de soluções de software avançadas para MCUs, como a implantação de modelos de aprendizagem profunda na borda. A sabedoria convencional sugere que hardware de computação cada vez mais poderoso é necessário e permite aplicações de software cada vez mais complexas. Nos últimos anos, no entanto, há incentivo para desenvolvermos soluções que permitam aos microcontroladores assumir tarefas tão complexas para concretizar visões de detecção distribuída e computação, onde precisamos encontrar um equilíbrio entre custo (que pode gerar onipresença), desempenho, e eficiência energética. Para dar um nome a esse incentivo:

podemos comparar a potência necessária para transmitir um fluxo de dados de algum controlador remoto para um uplink de servidor e descobrir que esse requisito de energia supera em muito a potência que precisaríamos para realizar uma análise semelhante a bordo. Além disso, existem questões éticas e políticas complicadas que envolvem a implantação de sensores distribuídos que capturam e depois partilham fluxos de informações possivelmente identificativas, em vez de relatórios desidentificados de retornos de modelos. Como tal, temos visto MCUs assumirem tarefas cada vez mais complexas através da implementação inteligente de software, juntamente com arquiteturas cada vez mais avançadas que permitem um processamento mais poderoso em pacotes ainda mais pequenos.

Capacidade de computação

Unidades microcontroladoras possuem arquiteturas de CPU variadas. Destes, os fabricantes de chips podem licenciar a arquitetura ARM Cortex M de 32 bits, fazendo modificações a partir deste ponto de partida comum para introduzir recursos e diferenciar seus produtos. Por exemplo, o Nano 33 BLE Sense chama um U-Blox NINA-B306, que é um módulo MCU-BLE independente, que integra o nRF52840 MCU da Nordic Semiconductor, que licenciou a arquitetura subjacente da ARM, já que o nRF52840 é um ARM Cortex-M4. Embora isso seja geralmente representativo do controlador moderno, é importante notar que há uma grande variação na capacidade de computação, com núcleos de processamento que chegam a 8 bits, trazendo relevância moderna em aplicativos simples.

Tal como acontece com outros hardwares de computação, um clock MCU mais rápido permite maior desempenho temporal, ao custo de diminuir a eficiência energética. As velocidades de clock típicas ficam entre 50 e 500 MHz. Por exemplo, o Nano 33 BLE Sense tem clock de 64 MHz. Dissemos deliberadamente 'relógio MCU' aqui, em vez de relógio da CPU, uma vez que o relógio central de um MCU permite funções além da CPU e chega a outros periféricos (ADCs, DACs, etc.) por meio de escalares, de modo que um relógio mais rápido pode não apenas permitir um processamento mais rápido, mas também criar espaço (largura de banda) para funções não relacionadas à computação. Por exemplo, embora um clock de CPU inferior a 0,5 GHz possa não parecer impressionante, isso pode se traduzir em intervalos de amostragem teóricos para conversão analógico-digital da ordem de dezenas de nanossegundos, o que é mais do que suficiente para a maioria dos fenômenos físicos de interesse. Além disso, velocidades mais baixas podem, na verdade, ser benéficas, de forma um tanto contraintuitiva, para cronometrar intervalos mais longos com precisão.

Em última análise, o melhor núcleo de processamento para a sua aplicação dependerá da complexidade da tarefa, dos seus requisitos de desempenho e da dinâmica temporal. Uma coisa pela qual os MCUs são conhecidos é seu papel em facilitar respostas em tempo real ou quase em tempo real a eventos por meio de interrupções e sistemas de eventos. A latência dessas respostas estará obviamente ligada às especificações de desempenho do controlador. Outra coisa pela qual os MCUs são conhecidos é permanecerem em grande parte adormecidos no ambiente até serem chamados. O clock dinâmico pode garantir que um MCU consuma o mínimo de corrente possível até que uma interrupção desperte o MCU deste modo de baixo

consumo de energia para responder a qualquer evento. Considere um controle remoto de TV que fica em casa enquanto você está no trabalho, que pode responder ao pressionar de um botão, ativar seu controlador e começar a emitir a sequência necessária de pulsos eletromagnéticos de seu transmissor IR. A capacidade de ajustar o relógio do sistema para reduzir o consumo de energia é um recurso técnico poderoso – trocadilho intencional.

Armazenamento e memória

Uma consideração importante a ser feita ao abordar a memória e o armazenamento do MCU pela primeira vez é que as escalas com as quais você está acostumado a trabalhar em computadores provavelmente não se aplicarão. Os microcontroladores utilizam o flash como uma opção não volátil para a memória do programa, de modo que o mesmo programa e inicialização ocorram a cada reinicialização. Normalmente, os MCUs apresentam cerca de 1 MB de memória de programa. Os microcontroladores recorrem à memória de acesso aleatório (RAM) para armazenar dados de trabalho, mas é claro que isso é volátil, de modo que os dados de um ciclo de energia não podem persistir no próximo. No contexto deste curso, é interessante considerar o tamanho dos vários modelos. Uma das principais restrições do TinyML é garantir que os MCUs, como hardware com recursos limitados, possam se ajustar ao modelo desejado em primeiro lugar. Modelos compactos de reconhecimento de fala, por exemplo, requerem 20-30 kB, enquanto modelos mais complexos, como aqueles necessários para detecção visual de recursos, requerem centenas de kilobytes, realmente no extremo do que a maioria dos MCUs fornece.

Queremos destacar aqui que algumas placas microcontroladoras fazem interface com cartões SD como forma de ampliar a capacidade de armazenamento de seu sistema, bem como de criar um registro não volátil de dados, para aplicações que assim o necessitem. Pelo que vale, EEPROM é outra forma de memória on-board ou in-chip que alguns aplicativos incorporados (cartões inteligentes, por exemplo) utilizam para armazenar dados programáveis e persistentes.

No final, você deve considerar a escala da sua aplicação e o papel que a volatilidade pode desempenhar na limitação ou habilitação da função necessária.