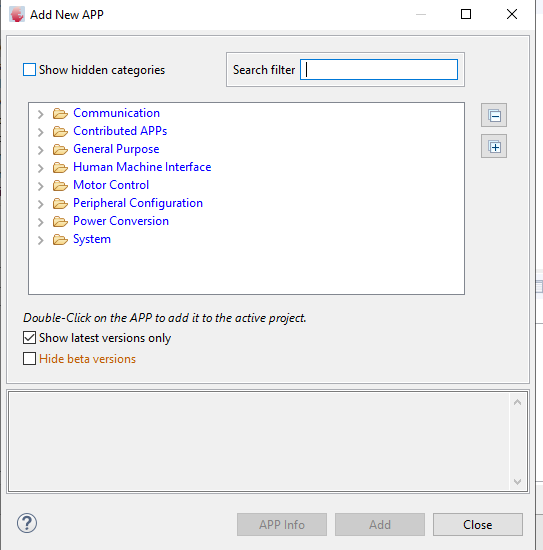
DAVE4 – APPs

O DAVE4 para facilitar a integração de algumas funções no microcontrolador possui uma camada de alto nível que permite adicionar as mesmas sem ser necessário o conhecimento de registos que de outra forma seria necessário. Algumas das funções mais usadas são: protocolos de comunicação, controlo de portas (*Outputs/Inputs)*, leitura de portas analógicas (*Analog-to-Digital Converter* - ADC*s*), geração de PWMs (*Pulse-width Modulation*), gerar interrupções, criar *timers*, entre outras.

Para adicionar uma APP ao nosso projeto apenas temos de a adicionar na tab “*App Dependency Tree*”*,* exemplificado na Figura 1 e 2.



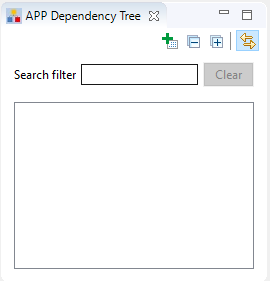




Figura – Aba de APPs

Figura 2 – APPs disponíveis

A título de exemplo, podemos adicionar um *Output* que irá controlar o LED1 disponível no módulo de desenvolvimento da Infineon, XMC4200 Platform2Go. Para tal devemos carregar na categoria “*System”* e adicionar a APP chamada “*DIGITAL\_IO*”. Após a adicionarmos devemos configurá-la, uma vez que esta vai controlar uma saída devemos configurar a mesma com *Output*, Figura 3.

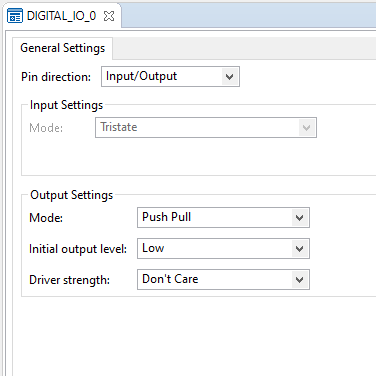


Figura 3 – Configuração da APP *Digital\_IO*

Em caso de dúvidas para configurar a APP, bem como conhecer todos os métodos já definidos com a APP, devemos abrir a documentação da mesma para explorar exemplos e explicações do funcionamento das mesmas. Para abrirmos a aba de ajuda devemos carregar com o botão direito do rato em cima da app e carregar em “*App Help”.*

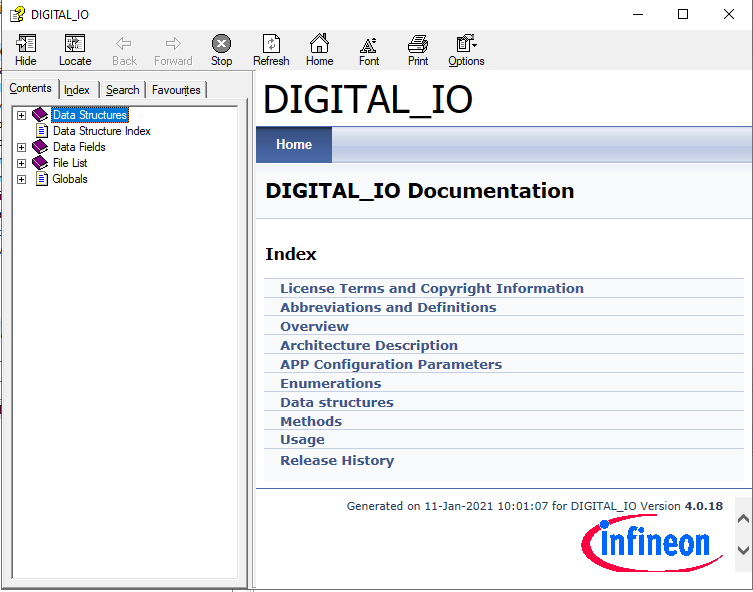


Figura 4 – Menu de ajuda da APP *Digital\_IO*

Após configurar a APP devemos alocar os pinos do microcontrolador que queremos usar, para isso devemos carregar na APP com o botão direito do rato e escolher “*Manual Pin Allocator*”. Na janela apenas temos de escolher o pino correto, que no caso do LED1 é P0.1.

Por fim devemos gerar o código da APP, para isso devemos carregar no ícone “*Generate code*”, Figura 5.

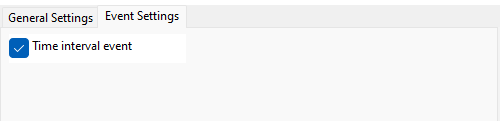
Figura 4 – Menu do DAVE

**Sugestão de pesquisa:** Porquê que definimos o *output* com modo *Push Pull* e não *Open Drain?*

Quais são as diferenças destes dois modos?

Para adicionar um pouco mais de complexidade a este pequeno exemplo, podemos alternar o estado deste led através do uso da *app* TIMER(4.1.12) e INTERRUPT(4.0.10). A ideia aqui é criar um timer que gera uma interrupção ao fim de um certo período de tempo. Desta forma podemos controlar facilmente a cadência com que o led pisca.

Depois de adicionar as duas *apps*, devemos passar à configuração das mesmas. Na configuração da app timer, devemos selecionar qual a periodicidade que desejamos e dar indicaçãode que queremos correr uma interrupção ao fim desse período.



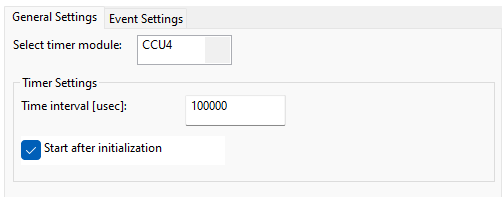


Figura 5 – Configuração da *app TIMER*

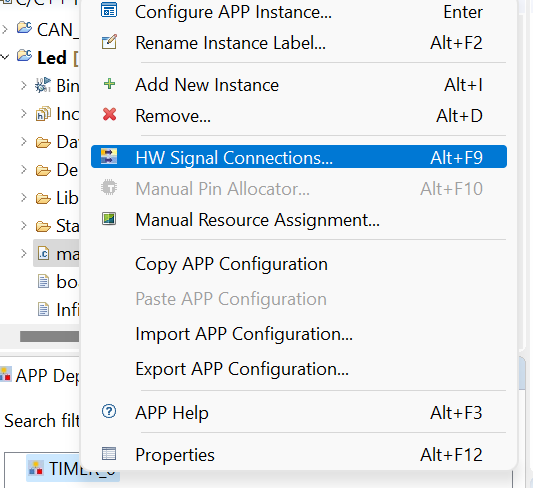
No entanto, é ainda necessária a criação de uma app *Interrupt*. Nesta fase irá definir-se o nome da função que será corrida quando a interrupção for acionada.

Uma imagem com texto

Descrição gerada automaticamente

Figura 5 – Configuração da *app Interrupt*

Adicionalmente será necessário conectar esta interrupt ao sinal que a irá espoletar, que neste caso é o sinal da app Timer. Portanto, teremos de ir ao menu de conexões físicas dos sinais do Timer e conectar esta interrupção.



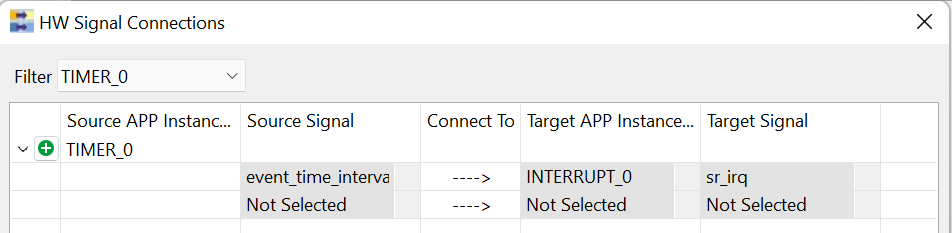


Figura 6 – Conexão física da *app* *interrupt* e *timer*

Por fim, será necessário o desenvolvimento da lógica dentro da função *interrupção\_periodica()* que irá desencadear a mudança do estado do led. Esta função terá de ser definida obrigatoriamente fora do *main().*

**Exercícios propostos:**

1. Colocar Led a piscar com uma cadência de 1s;
2. Utilizando apenas a interrupção conectada ao Timer, como ligar o led quando se clica num dos botões da placa? Desenvolva uma possível solução.