

Documentação do software do sistema embarcado - requisitos e modelagem

Modelo machine learning (TinyML) de reconhecimento de movimento usando Raspberry Pi Pico

Jonas Lendzion Schultz II

link GitHub:

<https://github.com/jonas-ii/jonasii-pico-motion.git>

Mestrando em Engenharia Elétrica
202304977
jonasschultz1309@gmail.com

1 Introdução

O presente relatório tem como objetivo documentar o software utilizado no sistema embarcado e no hospedeiro (PC), bem como os requisitos de modelagem do projeto final da disciplina EEL510265 - Programação de Sistemas Embarcados ministrada pelo Prof. Eduardo Augusto Bezerra.

Fazendo um breve resumo do projeto final, a ideia é implementar um modelo de machine learning embarcado (TinyML) em uma Raspberry Pi Pico para a identificação de movimentos.

Para se captar esses movimentos, utiliza-se um acelerômetro MMA845X. Diferentemente da proposição original, esse é um acelerômetro digital (e não analógico), logo, para o software embarcado na Raspberry Pi Pico, foi adicionalmente necessário implementar a comunicação via i2C aumentando um pouco mais a complexidade do projeto.

2 Coleta de dados - 1^a versão do software embarcado (sem TinyML)

A própria empresa Raspberry Pi possui um repositório no GitHub, no qual existe um exemplo de utilização de um acelerômetro MMA8451. Ao implementar o exemplo '.c' em conjunto com uma 'CMakeLists', percebeu-se que alterações são necessárias nesses arquivos com o intuito de proporcionar o funcionamento do acelerômetro digital em específico - com a árvore de projeto inicialmente proposta. Percebe-se que a maioria do exemplo está escrito em *C*, no entanto, esta versão é somente uma preliminar e a utilizada no projeto está em *C++* como requisitado.

Finalmente, a estrutura em árvore do arquivo do projeto de coleta de dados é a seguinte:

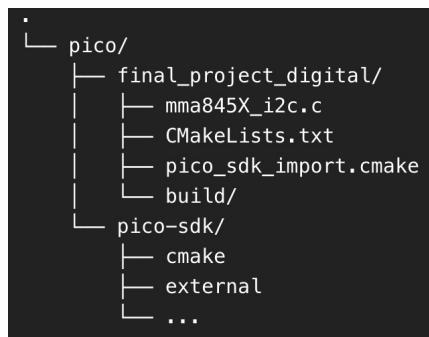


Figure 1: Árvore do projeto (sem ML)

Uma vez feito o 'cross compiler', envia-se o arquivo com a extensão '.uf2' para a Raspberry Pi Pico.

O set-up utilizado é o seguinte:

por uma interface serial para o estúdio. Este método só funciona perfeitamente em sensores com frequências de amostragem mais baixas, como no caso do projeto em questão (gestos humanos).

O encaminhador de dados é usado para retransmitir facilmente dados de qualquer dispositivo para o Edge Impulse via serial. Os dispositivos gravam os valores dos sensores por meio de uma conexão serial e o encaminhador de dados coleta os dados, assina-os e envia-os ao serviço de ingestão.

Dentro desse programa, pode-se plotar os dados (movimentos humanos realizados) para verificar se eles estão bem definidos e com isso será realizada separação do dataset em treino/validação e teste.

A imagem abaixo mostra a divisão dos clusters dentro da plataforma:



Figure 4: Divisão dos clusters - movimentos

Uma rede neural simples foi implementada para prever o movimento humano realizado. Ela possui 4 neurônios de saída - já que temos 4 classes possíveis de movimentos.

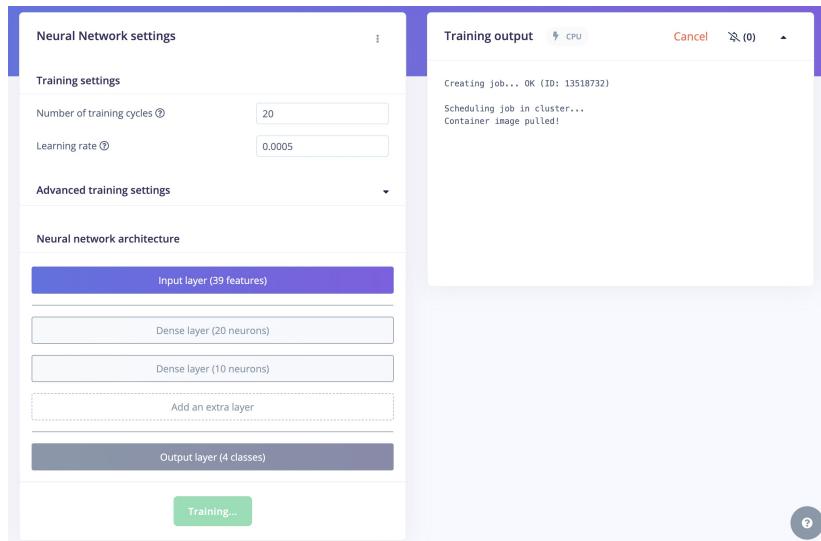


Figure 5: Rede neural implementada no treinamento do modelo

Após o seu treinamento, o seguinte desempenho foi alcançado:



Figure 6: Desempenho da rede neural com o conjunto de treinamento

E ao aplicar o conjunto de teste, o resultado obtido foi o seguinte:

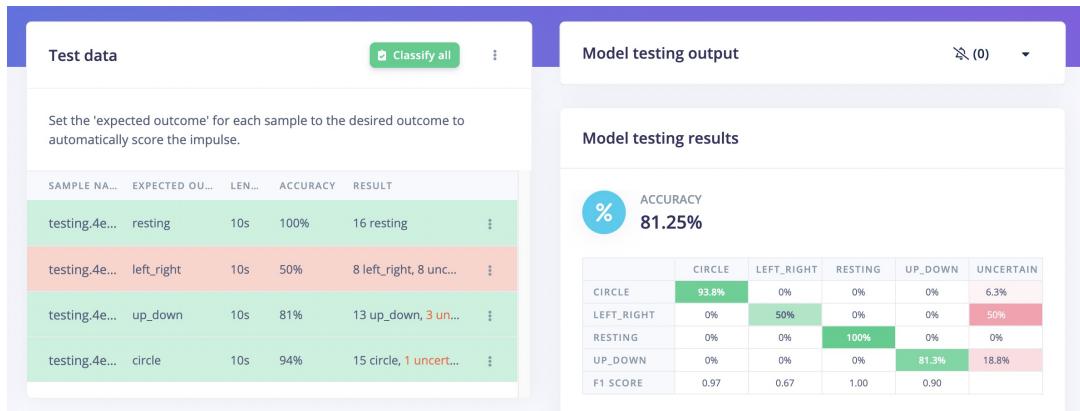


Figure 7: Desempenho com o conjunto de teste

Ressalta-se que os resultados obtidos poderiam ainda ser melhorados ao aumentar o conjunto de treinamento. No caso do projeto em questão, capturou-se 60 segundos de cada movimento para o conjunto de treinamento.

Ao final de toda a previsão e desenvolvimento do modelo, deve-se realizar a conversão e modificação dos arquivos gerados para o envio à Pico. O Edge Impulse possui uma seção de Deployment para auxiliar em que pode ser escolhida a extensão C++ para os arquivos.

O requisito principal dessas modificações é atender a proposição de utilizar uma fila de eventos (nesse caso um buffer com um ID específico para cada evento) dos gestos humanos e adicionar a função de leitura do rtc interno para a adição da data/hora do evento.

Finalmente, a estrutura final do projeto é a seguinte:

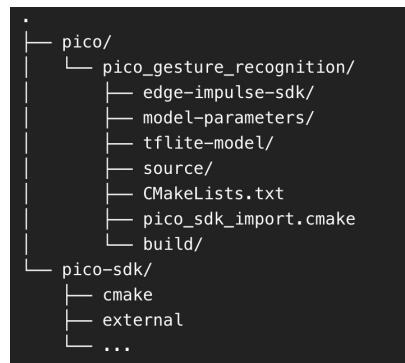
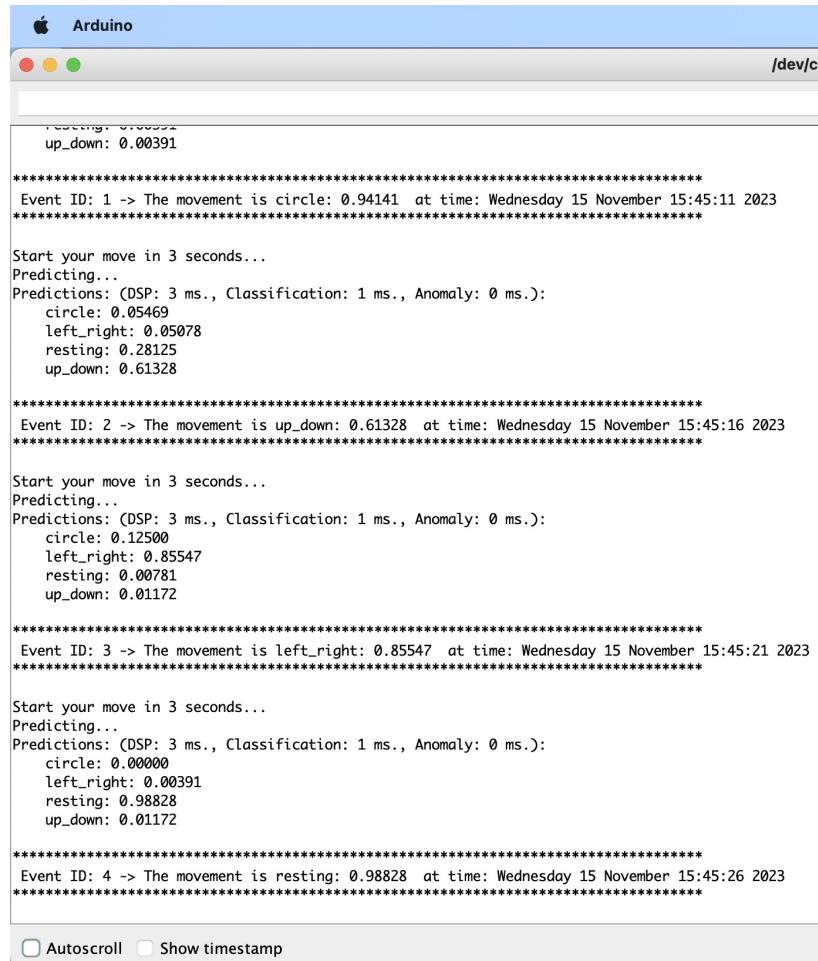


Figure 8: Árvore do projeto (com ML)

Após a criação de um arquivo 'main.cpp' dentro da pasta *sources/* e das respectivas mudanças para o envio da mensagem via USB/Serial, tem-se o seguinte resultado:



```

Arduino
/dev/cu

Reading: 0.00391
up_down: 0.00391
*****
Event ID: 1 -> The movement is circle: 0.94141 at time: Wednesday 15 November 15:45:11 2023
*****


Start your move in 3 seconds...
Predicting...
Predictions: (DSP: 3 ms., Classification: 1 ms., Anomaly: 0 ms.):
    circle: 0.05469
    left_right: 0.05078
    resting: 0.28125
    up_down: 0.61328

*****
Event ID: 2 -> The movement is up_down: 0.61328 at time: Wednesday 15 November 15:45:16 2023
*****


Start your move in 3 seconds...
Predicting...
Predictions: (DSP: 3 ms., Classification: 1 ms., Anomaly: 0 ms.):
    circle: 0.12500
    left_right: 0.85547
    resting: 0.00781
    up_down: 0.01172

*****
Event ID: 3 -> The movement is left_right: 0.85547 at time: Wednesday 15 November 15:45:21 2023
*****


Start your move in 3 seconds...
Predicting...
Predictions: (DSP: 3 ms., Classification: 1 ms., Anomaly: 0 ms.):
    circle: 0.00000
    left_right: 0.00391
    resting: 0.98828
    up_down: 0.01172

*****
Event ID: 4 -> The movement is resting: 0.98828 at time: Wednesday 15 November 15:45:26 2023
*****


 Autoscroll  Show timestamp

```

Figure 9: Serial Plotter do software embarcado

O evento, inicialmente presente em um buffer do sistema embarcado, é enviado via USB/Serial para o hospedeiro (já que, conforme relatado pelo professor, não há mais a necessidade de se usar uma UART). Nessa etapa utilizou-se somente o Serial Plotter da interface Arduino, porém, para a próxima seção, essa fila (buffer) será enviada para uma outra fila no hospedeiro de acordo com as especificações propostas.

Observa-se que esses eventos possuem um ID e uma data/hora específico. Após a validação dessa etapa, parte-se para o desenvolvimento do software do hospedeiro.

3.1 Software do hospedeiro (PC)

A primeira modificação foi feita no software embarcado na Pico, pois visa-se reduzir o tamanho da mensagem já que a mesma será copiada para um 'log' em forma de fila no host (PC).

Dante disso, somente a mensagem presente entre os asteriscos (*) da Figura 10 é enviada via USB/Serial para o host.

O host possui um sistema operacional MacOS, logo, foi utilizado o 'Xcode' em uma forma de *Xcode project* para a realização do programa em *C++*.

O programa acessa a porta USB correspondente à conexão da Raspberry Pi Pico - consequentemente também seta a *Baud Rate* necessária e o caminho até essa porta - e define as funções correspondentes para realizar a leitura dos eventos enviados pelo setup embarcado. Cabe, então, ao usuário escolher entre as diversas opções disponibilizadas o que ele deseja executar, conforme a imagem abaixo:

```
*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option:
```

Figure 10: Menu do Software do Host (PC)

Alguns casos do funcionamento desse software do host serão mostrados em seguida.

Inicialmente, mostra-se a funcionalidade de adquirir um evento (movimento) do sistema embarcado e adicioná-lo a uma fila, além disso, como requisitado, é possível ao 'admin' visualizar os eventos presentes nessa fila:

```
*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 1

The event is: Event ID: 444 -> The movement is resting: 0.97266 at time:
Wednesday 15 November 16:22:12 2023

*****  

Welcome 'admin'  

Press 1 if you want to read a movement (5 seconds)  

Press 2 if you want to check the log (queue)  

Press 3 if you want to clear the log (queue)  

Press 4 if you want to see the active time of Pico and close the serial port  

*****  

Option: 2

The event 0 in the queue of the host is:  

Event ID: 411 -> The movement is resting: 0.97266 at time:  

Wednesday 15 November 16:19:26 2023  

The event 1 in the queue of the host is:  

Event ID: 430 -> The movement is resting: 0.98047 at time:  

Wednesday 15 November 16:21:01 2023  

The event 2 in the queue of the host is:  

Event ID: 432 -> The movement is resting: 0.98438 at time:  

Wednesday 15 November 16:21:11 2023  

The event 3 in the queue of the host is:  

Event ID: 444 -> The movement is resting: 0.97266 at time:  

Wednesday 15 November 16:22:12 2023

*****  

Welcome 'admin'  

Press 1 if you want to read a movement (5 seconds)  

Press 2 if you want to check the log (queue)  

Press 3 if you want to clear the log (queue)  

Press 4 if you want to see the active time of Pico and close the serial port  

*****  

Option: 3

The queue is now empty
```

Figure 11: Caso de funcionamento do software do host para adicionar eventos na fila e visualizá-la

Em seguida, mostra-se que o usuário 'admin' também pode reiniciar a fila ao apagar todos os eventos previamente listados nela:

```
*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 2

The event 0 in the queue of the host is:
Event ID: 472 -> The movement is resting: 0.95703 at time:
Wednesday 15 November 16:24:32 2023
The event 1 in the queue of the host is:
Event ID: 475 -> The movement is resting: 0.98047 at time:
Wednesday 15 November 16:24:47 2023
The event 2 in the queue of the host is:
Event ID: 484 -> The movement is resting: 0.97656 at time:
Wednesday 15 November 16:25:32 2023

*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 3

The queue is now empty

*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 2

*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
```

Figure 12: Caso de funcionamento do software do host ao esvaziar a fila

Além disso, conforme a especificação do projeto, o usuário 'admin' também pode acessar o tempo em que a Raspberry Pi Pico esteve ativa, nesse caso, o tempo é dado em μ s.

```
*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 1

The event is: Event ID: 502 -> The movement is resting: 0.91016 at time:
Wednesday 15 November 16:27:02 2023

*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 1

The event is: Event ID: 504 -> The movement is resting: 0.95703 at time:
Wednesday 15 November 16:27:12 2023

*****
Welcome 'admin'
Press 1 if you want to read a movement (5 seconds)
Press 2 if you want to check the log (queue)
Press 3 if you want to clear the log (queue)
Press 4 if you want to see the active time of Pico and close the serial port
*****
Option: 4

The active time of Pico is: 2537015234 (us)

Serial port closed

Program ended with exit code: 0
```

Figure 13: Caso de funcionamento do software do host ao requisitar o tempo em que o micro-controlador esteve ativo

Finalmente, após as explicações do funcionamento tanto do software embarcado e o software do host (PC), tem-se o seguinte diagrama de classes:

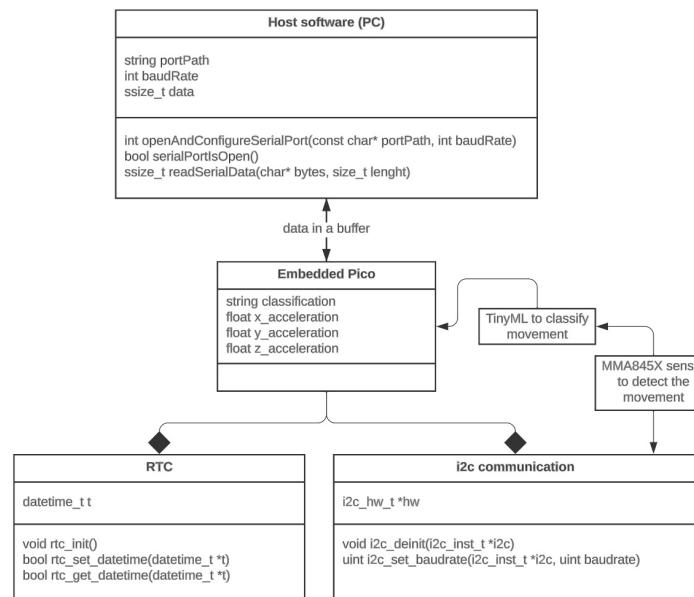


Figure 14: Diagrama de classes simplificado do projeto