

TÉCNICO SOLAR BOAT

Projetos de Recrutamento

Sistemas Elétricos

2021/2022 - 1º Semestre

I Introdução

O objetivo deste documento é enunciar diferentes projetos a serem realizados no âmbito do recrutamento do projeto Técnico Solar Boat na área de Sistemas Elétricos. De entre estes projetos deves escolher um, realizar todas as alíneas (exceto no primeiro projeto onde te apresentamos 2 opções) e apresentar a melhor solução que conseguires, todo o teu desenvolvimento irá ser acompanhado e caso tenhas qualquer dúvida, não hesites em contactar-nos! Apresentamos-te vários projetos diferentes de diversos temas: Barco Autónomo, *AI Vision*, Programação em C++ e Website. Escolhe um dos vários projetos para fazeres.

Na drive onde te disponibilizamos este documento existe uma pasta com vários anexos que também poderás usar como apoio ao teu trabalho.

Data Limite de Entrega: 7 de Novembro de 2021

Submissão por email - Assunto: "Projeto de recrutamento SE: (nome do projeto/exercício) - (1º e último nome)"

A área de sistemas eléctricos é responsável pela geração (painéis fotovoltaicos), armazenamento (bateria) e consumo (cargas, principalmente o motor) de energia elétrica nos barcos. Para além destes três pilares existe uma gama de sensores e atuadores nos barcos que tratam da segurança do sistema elétrico. Recentemente a área é também responsável pelo controlo dos foils, permitindo assim que o casco do barco sobrevoe a água, obtendo uma melhor eficiência. Somos também responsáveis por desenhar e implementar todo o sistema elétrico de ambos os nossos protótipos, o SR03, movido a energia solar e o SM01 movido a hidrogénio. Toda a informação de ambos os protótipos também é transmitida para um servidor onde a equipa em terra pode monitorizar o estado do barco em tempo real.



Figura I.1: SR03 no Mónaco

II Projectos de Recrutamento

A nossa equipa preparou três projetos entre os quais podes escolher o que achares mais interessante. Todos os documentos relacionados com os projetos podem ser encontrados na drive na pasta **Media**.

II.1 Barco Autónomo

O TSB, enquanto projeto, procura estar sempre na linha da frente da inovação tecnológica, procurando mais e melhores soluções para veículos sustentáveis, à medida que vai desenvolvendo os seus protótipos. O projeto do barco autónomo encaixa perfeitamente nestes critérios, apresentando-se com um novo desafio entusiasmante e de enorme potencial.

No final do ano passado, participámos no Njord Challenge, uma competição inteiramente online que nos permitiu ter um primeiro contacto com o que um sistema autónomo requer. Assim, decidimos dividir a área do barco autónomo em visão e controlo, que se encontram interligadas pelo ROS (Robot Operating System).

Na competição, desenvolvemos as tarefas e testámos o sistema utilizando um simulador desenvolvido por eles, que podes encontrar no seguinte link: <https://njord.gitbook.io/njord/>. O website contém informação sobre como instalar o simulador, bem como todos os sensores que este incorpora.

Tencionamos continuar a desenvolver o nosso sistema e a testá-lo no simulador, pelo que será necessário instalá-lo para realizar o recrutamento (qualquer problema com a instalação não hesites em contactar-nos para te podermos ajudar).

Apresentamos em baixo dois projetos alternativos, onde podes escolher aquele que se enquadra melhor com os teus conhecimentos/interesses.

II.1.a Projeto 1 - Path Planning

Neste projeto o objetivo passa por aplicar com sucesso um algoritmo de path planning no ROS. Para isto, podes optar entre desenvolver o teu próprio algoritmo ou implementar uma package já existente, fazendo todos os ajustes de compatibilidade. O algoritmo deve ser capaz de:

- com base num mapa que iremos disponibilizar na drive, que nos foi fornecido na competição, definir a melhor trajetória para o barco atingir o ponto gps pretendido. O ponto gps é obtido subscrevendo a um tópico que contém a informação da posição final pretendida; (caso notes que há alguma tipo de incoerência entre as coordenadas do mapa e as fornecidas no tópico do ROS, contacta-nos)
- ir ajustando a trajetória com base nos dados do sensor de distância (LiDAR), de forma a desviar-se de obstáculos estacionários.

O output final deverá ser um conjunto de velocidades para fornecer ao nosso controlador, que irá depois converter nas forças necessárias de aplicar à embarcação.

Numa fase mais avançada, iremos trabalhar para incorporar obstáculos em movimento no planeamento da trajetória, bem como incorporar as informações da visão no algoritmo.

Na competição, a solução que utilizámos foi desenvolver um controlador de posição, que determinava a trajetória da embarcação. Esta solução é longe de ótima, e existem packages muito mais eficientes com o mesmo propósito.

NOTA: Este projeto é de nível avançado, e é preferível para alguém já com experiência em ROS.

II.1.b Projeto 2 - Fusão AI-Lidar

Após a AI reconhecer um objeto, é necessário proceder à caracterização do mesmo para fornecer a sua posição ao controlador. Assim, neste projeto, o objetivo passa por fundir os dados do sensor de distância (LiDAR), com a informação proveniente da AI encontrar um obstáculo.

És livre de implementar a tua solução como achares mais conveniente, mas aqui fica a nossa sugestão: a AI fornece informação sobre os pixels onde está presente um obstáculo, pelo que é possível determinar o ângulo em relação ao barco onde se encontra este obstáculo, sendo por isso possível fazer uma correspondência com os pontos do LiDAR nesse ângulo. No futuro, pensando já num modelo físico, poderá também ser necessário fazer as transformações de referenciais necessárias para os dados de ambos os sensores coincidirem (calibração dos sensores).

Também existem inúmeros packages do ROS com esta finalidade, pelo que também podes implementar uma delas no teu projeto, assegurando todas as condições de compatibilidade.

NOTA: a tua solução deve ser implementada utilizando o ROS, provavelmente criando um nó dedicado a esta conversão, e subscrevendo às mensagens publicadas pelo simulador.

II.2 AI Vision

No ano passado participámos pela primeira vez numa competição que consistia em desenvolver software para um barco navegar autonomamente. A competição foi somente online e foi-nos fornecido um software onde era simulado um barco com os seguintes sensores: 4 câmeras, LiDAR e GPS. Uma das provas da competição consistia em navegar num canal seguindo as bóias e evitando colisões com outras embarcações. Sendo assim necessário reconhecer as cores das bóias e a localização das mesmas. Para isso treinámos uma rede neuronal com imagens que recolhemos.

O objetivo deste projeto é treinar uma rede com imagens desse simulador que estão disponíveis na pasta da drive.

II.2.a Labelling

Na drive está um conjunto de imagens que foram recolhidas do simulador. Caso necessites de mais imagens, fala connosco e nós ajudamos-te. Queremos que a rede neuronal consiga identificar 4 objetos/obstáculos:

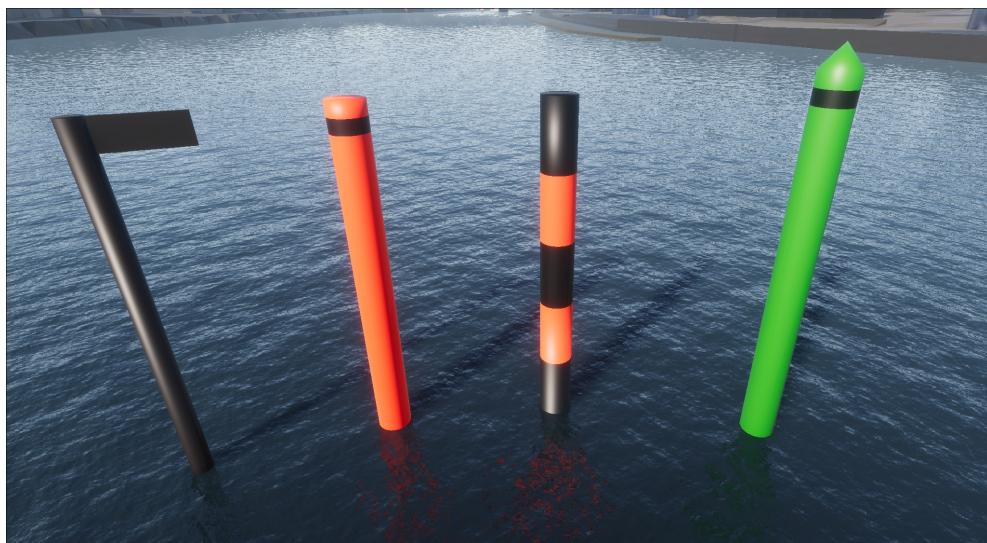


Figura II.1: Objetos a identificar

Da esquerda para a direita, estes objetos são:

- Marca de águas pouco profundas, a bandeira aponta sempre para águas profundas.
- Marca lateral vermelha
- Marca de perigo isolado
- Marca lateral verde

Podes usar qualquer software para fazer label das imagens, existem vários gratuitos disponíveis online.

II.2.b Training

Relativamente ao desenvolvimento e treino da rede neuronal capaz de fazer a deteção dos objetos, recomendamos a utilização de tensorflow ou de PyTorch (bibliotecas para machine learning que têm uma série de funções que facilitam imenso o desenvolvimento e treino das redes) e o google colabatory, que consiste num editor de código do browser para python que permite a utilização de GPU remotos gratuitamente e com capacidade suficiente para treinar redes neurais, ao contrário da maioria dos computadores comuns. A ideia passa por desenvolver uma rede capaz de conseguir detetar obstáculos com uma alta eficácia, por isso, na eventualidade de

considerares o dataset de imagens fornecido curto para o atingir, disponibilizaremos mais imagens do simulador com os marcadores náuticos. Numa fase mais avançada do projeto de recrutamento, seria também importante que conseguisses exportar a rede neuronal treinada e a implementasses num nó do ROS, que iria receber frames com uma determinada refresh rate, detetando todos os obstáculos associados a cada frame.

II.3 Programação em C++

Ambos os nossos protótipos contêm várias PCBs únicas que necessitam de código especialmente concebido para realizar a sua função. Na sua maior parte, este código é escrito em C++. Recorremos a algumas bibliotecas para simplificar o código, sendo que algumas delas são feitas por nós. Os micro-controladores que utilizamos nestas PCBs são os Teensy e os AT90CAN. Todos estes sistemas comunicam entre si pelo protocolo de comunicação CAN-BUS, que é o protocolo standard hoje em dia em qualquer tipo de veículo, seja ele um barco ou um avião. Além disso alguns destes sistemas têm ainda de recorrer a outros protocolos como por exemplo RS232 ou RS485 para conseguirem adquirir dados de diferentes sensores que temos no barco.

Queremos que desenvolas um software simples que lê dados de um sensor que comunica através de **RS485** e envie esses dados para o CAN-BUS do barco.

Há várias maneiras de programar os nossos micro-controladores, no entanto, aconselhamos-te a utilizar o Visual Studio Code e a extensão PlatformIO. Após instalares a extensão cria um projeto novo dentro da janela do PlatformIO e escolhe como board o Teensy 3.5 e como framework Arduino.

II.3.a Investigar o Hardware

Irás utilizar um Teensy 3.5 para ler e enviar os dados. Queremos que faças um esquema elétrico das ligações e dos componentes que irás necessitar para conseguir obter os dados do sensor e posteriormente enviá-los para o CAN-BUS do barco. Este esquema não tem de ser nada muito elaborado, o importante é que nós consigamos perceber que tu sabes o que precisas e como fazer as respetivas ligações.

II.3.b Ler Dados do Sensor Ultrassom

Agora que já sabes que hardware necessitas escreve o código necessário para ler os dados enviados pelo sensor ultrassom do barco. Este sensor é utilizado para calcular a distância do barco à água e como tal é um componente fulcral do sistema de controlo dos foils. O sensor que utilizamos é o TSPC-30S1-485 da Senix. A datasheet do sensor está disponível na pasta da drive.

Nós já temos uma PCB que vais poder usar para testar o teu código, pelo assim assim que tenhas algo pronto a testar fala connosco para podermos ajudar-te com essa parte.

II.3.c Enviar os dados por CAN

Depois de conseguires ler os dados do sensor é agora necessário enviá-los por CAN para o resto dos sistemas terem acesso a eles. No barco utilizamos um baudrate de 1 Mbits/s. O protocolo que utilizamos é o seguinte:

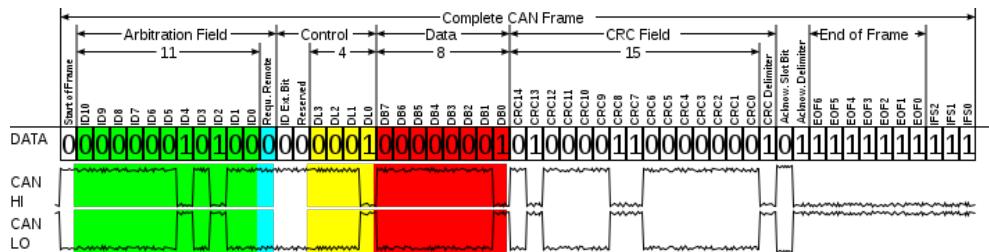


Figura II.2: Estrutura das mensagens

Estrutura do ID

Priority bits	Data ID	Source ID
2 bits	5 bits	4 bits

Priority bits

- Serve para definir a prioridade das mensagens.
- 00 prioridade máxima
- 11 prioridade mínima

Data ID

- Define a mensagem a ser transmitida por cada dispositivo
- Relativo a cada Source ID

Source ID

- Define qual o Node enviou a mensagem
- Juntamente com o Data ID determina qual a mensagem foi enviada

O ID da mensagem com os dados do sensor é à tua escolha desde que respeite o nosso protocolo.

Para enviar as mensagens para o CAN-BUS podes usar as funções da biblioteca Buffer que te disponibilizamos. Coloca-a na pasta lib do teu projeto de PlatformIO.

Recomendamos que utilizes a biblioteca [FlexCAN_T4](#) para interagir com hardware de CAN do Teensy uma vez que esta é a que usamos no projeto.

II.4 Website

O projeto conta com 2 websites, um de acesso geral acessível através deste [Link](#), e um website que utilizamos para gerir o projeto e que nos ajuda a manter tudo organizado.

O primeiro website está escrito em HTML, JavaScript e PHP, já o segundo foi feito recentemente e usa React.js com Typescript. Estamos neste momento a desenvolver um novo website principal utilizando estas novas linguagens. No futuro precisaremos de alguém que faça manutenção e implementação de novas *features* nos websites, isto seria parte do teu trabalho.

II.4.a Dinosaur Game

Penso que conheces o Dinosaur Game do Chrome, é um jogo simples que é acessível quando não tens ligação à Internet. O objetivo deste projeto é fazeres uma réplica desse jogo com React.js e Typescript. Podes utilizar outras bibliotecas existentes se assim entenderes no entanto, o uso de bibliotecas que implementem o jogo completamente não é permitido. O código deverá estar comentado e organizado.

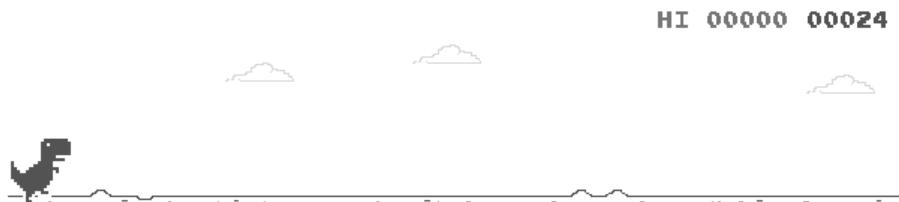


Figura II.3: Dinosaur Game

Para dar host ao website podes utilizar o serviço de Página Web que os Serviços de Informática do técnico oferecem. Podes saber mais [aqui](#) e para ativar os serviços [clica aqui](#) (tens de ativar o AFS, Shell, Web e CGI). Para carregar os ficheiros para a tua página pessoal segue [estas indicações](#) e cola os ficheiros na pasta "web". Alguma dúvida podes sempre contactar-nos e nós ajudamos-te.

III Passo Final - OBRIGATÓRIO

Por fim, **após teres submetido o teu projeto (só após!!)**, pedimos-te que preenchas este [Forms](#) que serve para nós ficarmos a saber como conheceste o projeto, bem como mais um pouco sobre ti!

Boa sorte nesta fase de recrutamento ;)