**Simtwo:**

Usando rede wifi:

Config -> I/O -> check em “Listen” (porta 9808)

Sheets -> Click no botão “UDP”

-> (20,2) = IP do Pi Pico

Se não aparecer o IP do Robô. Mudar no código do Pi Pico, ficheiro “main.cpp” linha 370:

void loop()

{

//if (WiFi.connected() && !ip\_on) {

if (WiFi.connected()) {

…

Verificar depois na porta série o IP assumido pelo Pi Pico e colocar na célula (1,2) do SimTwo.

-> Click no botão “Robot” para posicionar o robô (várias vezes até a estimativa Xe, Ye Thetae estar correta)

-> Quando for para iniciar o controlo, click no botão “Set” junto ao “1” (linha 4) para colocar no estado 1.

Usando Porta série:

Config -> I/O -> Com Port

Selecionar a porta série e clicar em “Open”

Sheets -> Click no botão “Serial”

-> Click no botão “Robot” para posicionar o robô (várias vezes até a estimativa Xe, Ye Thetae estar correta)

-> Quando for para iniciar o controlo, click no botão “Set” junto ao “1” (linha 4) para colocar no estado 1.

Sempre que se programar o micro através do Visual Studio Code tem que se fazer “Close” da porta no Simtwo. Depois de programado, o terminal da porta série no Visual Studio Code deve estar fechado e voltar a fazer “Open” no SimTwo.

**Como comunicam e se sincronizam os ciclos entre o Simtwo e o PI Pico:**

SimTwo-> “Procedure TimerEvent” é chamada da “procedure Control” em cada ciclo de controlo.

Esta chama “function PrepareData” que constrói a string com a informação a enviar por UDP ou porta série (State, enc1, enc2, sens\_line[.], touch\_sensor, ..) e o comando “go”. No código do Pi Pico, ficheiro “main.cpp”, função “void process\_command(command\_frame\_t frame)” recebe os dados do Sim Two e o comando “go” coloca “robot.control\_event = true”

Pi Pico-> main.cpp -> “void loop()”

if (robot.control\_event) -> faz o controlo do robô (chamando “control(robot)”, as máquinas de estados (“class main\_fsm\_t: public state\_machine\_t“ e outras que existam) e o processamento dos dados recebidos do Simtwo (odometry, localization, …), calculado o controlo envia o mesmo para o SimTwo através de comandos do tipo “serial\_commands.send\_command("u1", robot.u1);” O Simtwo recebe os dados através de procedure process\_char(var GChannels: TGChannels; b: char) e depois procedure process\_command(channel: TChannel).

**Controlo do Robô:**

Pi Pico-> control.cpp -> “class main\_fsm\_t: public state\_machine\_t”

A máquina de estados:

“class main\_fsm\_t: public state\_machine\_t”

trata do controlo do robô.

Pi Pico-> robot.cpp -> “robot\_t::robot\_t()”

Ajustar os valores de

“follow\_k” e “follow\_v”

**Ligação por wireless do PI Pico:**

Pi Pico-> main.cpp -> “void setup()”

Na linha 344 mudar o SSID e password para a rede wireless a utilizar.

**Visual Studio Code**

Ver em:

https://github.com/P33a/LAB0\_3

Essencialmente a parte:

Open Regedit

Go to HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem in regedit and then set LongPathsEnabled to 1.

Install Git

Open command line (CMD), run the command git config --global core.longpaths true

Reboot

Install VSCode