Relatório do Trabalho Final

PROGRAMAÇÃO DE SITEMAS EMBEBIDOS

RICARDO MACHADO – A22002860

GUILHERME SILVA – A22002696

Índice

1. Introdução

Este trabalho de programação de sistemas embebidos tem como objetivo utilizar uma câmara PS Eye para, quando é detetado movimento, captura fotografias que são posteriormente enviadas para um bot na aplicação de mensagens Telegram. O sistema embebido, constituído por um Raspeberry PI 4 e pela câmara, é responsável por realizar o processo de captura de imagem e enviar os dados para o bot, tornando possível a monitorização remota através do Telegram.

Para implementar esta solução, é necessário desenvolver um programa de controlo que interfira com a câmara PS Eye e capture imagens quando é detetado movimento. Esse programa também deve ser capaz de estabelecer uma conexão com o bot do Telegram para enviar as imagens capturadas em tempo real.

2. Materiais

- **Raspberry Pi 4** - é uma placa de desenvolvimento de computador de placa única criada pela Fundação Raspberry Pi. Foi concebida para fornecer uma solução compacta e acessível para o desenvolvimento de projetos eletrónicos e computacionais. A placa Raspberry Pi 4 é baseada num processador ARM Cortex-A72 quad-core de 64 bits, oferecendo um desempenho significativamente superior ao dos seus antecessores. Está disponível em diferentes configurações de RAM, que vão de 2 GB a 8 GB, permitindo a escolha adequada para satisfazer as necessidades específicas de um projeto. Além disso, o Raspberry Pi 4 possui uma variedade de funcionalidades e conectividade, incluindo portas USB 3.0, portas Ethernet, conectividade Wi-Fi e Bluetooth integrada, conectores HDMI para saída de vídeo e entrada de alimentação USB-C

Figura 1 - Raspberry Pi 4

- **Fonte de Alimentação USB-C Oficial do Raspberry Pi** - é um adaptador de energia concebido especificamente para utilização com dispositivos Raspberry Pi, incluindo o Raspberry Pi 4. É um produto oficial lançado pela Raspberry Pi Foundation, garantindo compatibilidade e desempenho ideal. A fonte de alimentação fornece uma fonte de energia estável e fiável para o Raspberry Pi, permitindo-lhe funcionar de forma suave e eficiente. Possui um conector USB-C, que é a interface de entrada de energia padrão para o Raspberry Pi 4. Este conector assegura uma ligação segura e fiável entre a fonte de alimentação e o Raspberry Pi. A fonte de alimentação USB-C oficial fornece uma saída de 5V e suporta uma corrente máxima de 3 amperes, o que é suficiente para satisfazer os requisitos de energia do Raspberry Pi 4 e dos seus periféricos ligados.

Figura 2 - Fonte de Alimentação USB-C Oficial do Raspberry Pi

- **Câmera PS Eye** - é uma câmara desenvolvida pela Sony para ser utilizada com as consolas PlayStation, como a PlayStation 3 e a PlayStation 4. Foi originalmente concebida para suportar a interação de movimentos e funcionalidades de jogo, como jogos baseados em movimentos, controlo por gestos e reconhecimento facial. A câmara PS Eye possui uma lente de alta resolução e é capaz de captar vídeo e áudio. Tem um sensor de imagem CMOS e suporta uma velocidade de fotogramas até 60 fps (fotogramas por segundo) com uma resolução de 640x480 pixéis. Além disso, a câmara pode captar vídeo de alta-definição (HD) com uma resolução de 1280x720 pixels a 60 fps. O PS Eye também possui capacidades de deteção de movimento, como o seguimento preciso dos movimentos e gestos dos jogadores. Pode ser utilizado para seguir a posição e os movimentos dos jogadores em frente à câmara, permitindo a interação com jogos e aplicações através de gestos e movimentos corporais. Embora tenha sido concebida principalmente para utilização com as consolas PlayStation, a câmara PS Eye também pode ser utilizada noutros contextos e projetos.

Figura 3 – Câmera PS Eye

- **Cartão microSD de 32 GB** - é um pequeno dispositivo de armazenamento concebido para ser utilizado em dispositivos que suportam cartões microSD, tais como smartphones, tablets, câmaras, consolas de jogos e outros dispositivos eletrónicos.

Figura 4 – Adaptador e Cartão microSD 32GB

- **Cabo HDMI com adaptador micro HDMI** - é uma combinação de um cabo com duas entradas HDMI e um adaptador micro HDMI que permite ligar dispositivos com uma porta micro HDMI a uma porta HDMI padrão. O adaptador micro HDMI é um pequeno conector que converte a porta micro HDMI, normalmente encontrada em dispositivos portáteis como smartphones, tablets ou câmaras, numa porta HDMI padrão. Isto permite-lhe ligar estes dispositivos a outros dispositivos que tenham uma porta HDMI, como televisores ou monitores.

Figura 5 - Cabo HDMI com adaptador micro HDMI

Os restantes materiais são para o uso após a configuração do Raspberry Pi 4:

- **Rato**
- **Teclado**
- **Monitor**

3. Implementação

3.1. Configuração do Raspberry Pi 4

É necessário fazer download do Raspberry Pi Imager para facilitar o processo de instalação do sistema operativo no cartão microSD.

É necessário também fazer download do sistema operativo desejado, que neste caso é o Ubuntu. Após fazer o download dos dois ficheiros, devemos inserir o cartão microSD no computador, com o auxílio a um adaptador. Depois de abrir o Raspberry Pi Imager, na opção "CHOOSE OS" é para selecionar o .zip do sistema operativo descarregado e no "CHOOSE STORAGE" é para selecionar o cartão microSD. Depois de selecionados, clica-se no "WRITE".

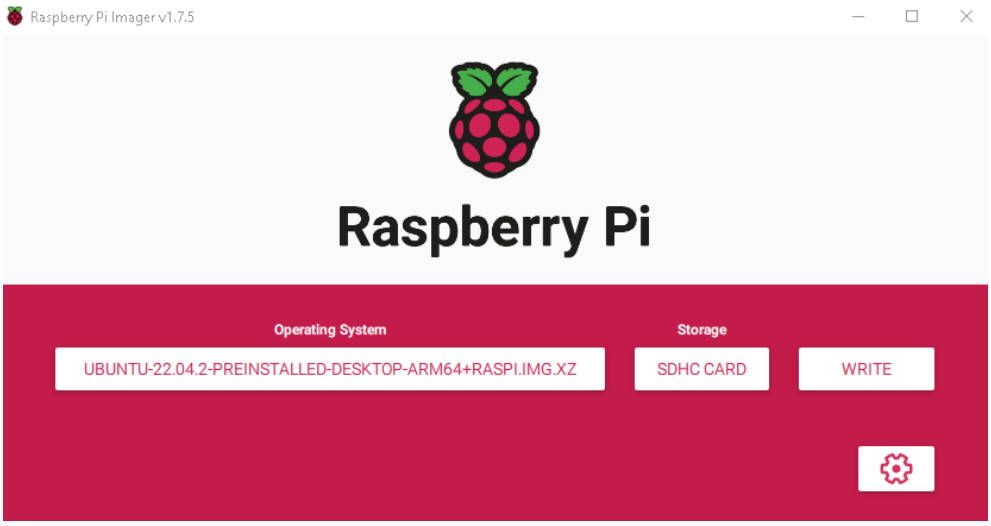


Figura 6 - Raspberry Pi Imager

Depois de o processo estar completo, deve-se inserir o cartão microSD e os periféricos (monitor, rato e teclado) no Raspberry Pi 4 para iniciar as configuração do sistema operativo.

Figura 7 - Raspberry Pi 4 com os componentes ligados

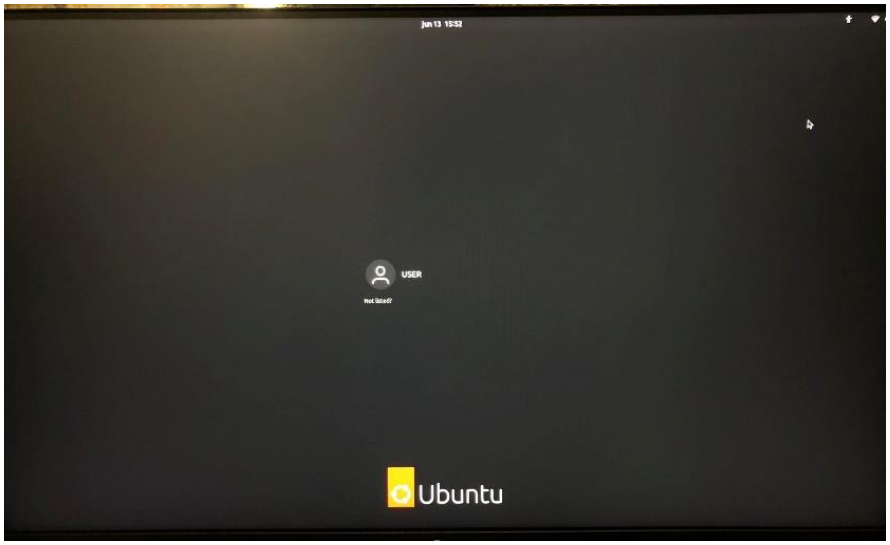


Figura 8 - Página de login

3.2. Configuração do Bot no Telegram

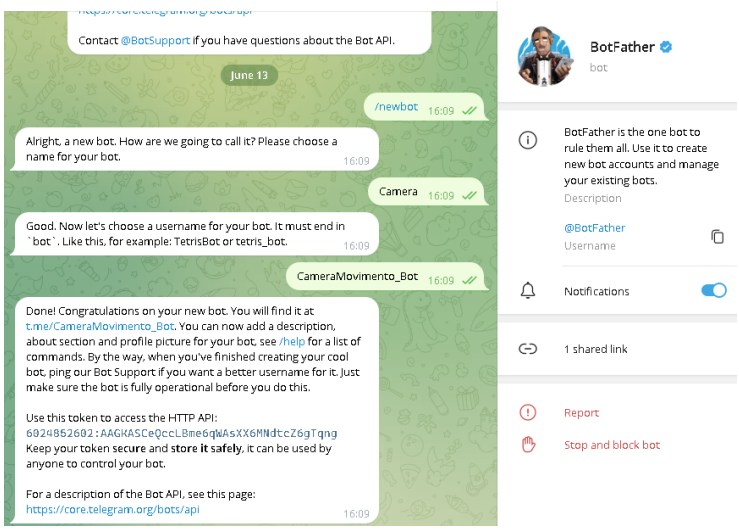


Figura 9 - Criação do CameraMovimento_Bot

O BotFather, ao criar o bot, dá automaticamente o bot_token, que será necessário no desenvolvimento do código e do bom funcionamento do projeto.

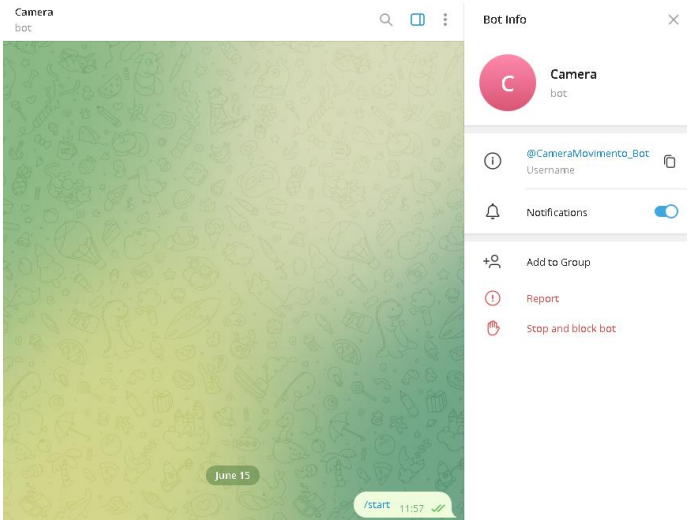


Figura 10 - CameraMovimento_Bot

Assim como o bot_token, o chat_id também é necessário para determinar o destinatário das fotos.

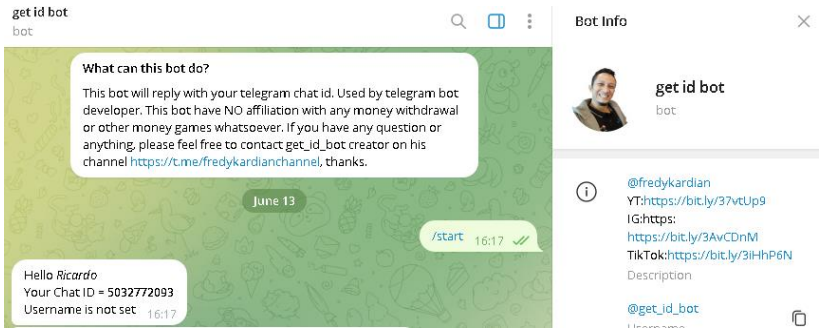


Figura 11 - get_id_bot

4. Desenvolvimento

4.1. Inicialização

Na consola (Terminal) do Ubuntu é necessário instalar algumas bibliotecas e ferramentas para controlar a câmera e o envio de fotos para o bot do Telegram:

- **sudo apt-get update**
- **sudo apt-get install python3-pip**
- **sudo pip3 install python-telegram-bot opencv-python**

4.2. Código

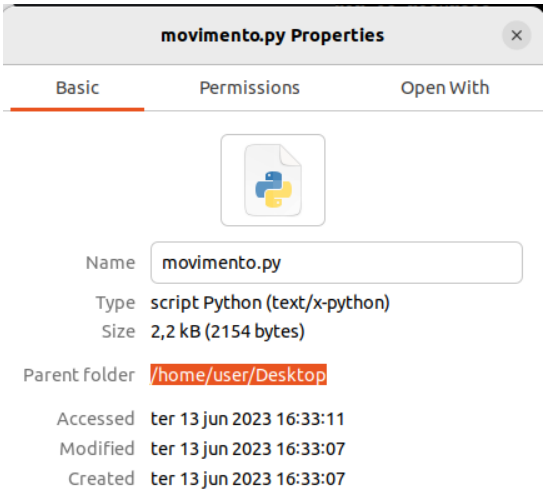
O código foi desenvolvido no **Text Editor**, mas foi guardado como ***.py***.

```python
import cv2
from telegram import Bot
from datetime import datetime
import asyncio

# Configurações do Telegram
bot_token = '6024852602:AAGKASCeQccLBme6qWAsXX6MNdtcZ6gTqng'
chat_id = '5032772093'

# Função assíncrona para enviar a foto
async def send_photo_async(bot, chat_id, photo):
    await bot.send_photo(chat_id=chat_id, photo=photo)

# Função principal assíncrona
async def main():
    # Inicializar o bot
    bot = Bot(token=bot_token)

    # Inicializar a câmera
    cap = cv2.VideoCapture(0)

    # Variável para detectar o primeiro frame
    first_frame = None

    while True:
        ret, frame = cap.read()

        if not ret:
            break

        # Converter para escala de cinza
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        gray = cv2.GaussianBlur(gray, (21, 21), 0)

        # Atualizar o primeiro frame
        if first_frame is None:
            first_frame = gray
            continue

        # Calcular a diferença entre os frames
        frame_delta = cv2.absdiff(first_frame, gray)
        thresh = cv2.threshold(frame_delta, 30, 255, cv2.THRESH_BINARY)[1]
        thresh = cv2.dilate(thresh, None, iterations=2)

        # Encontrar os contornos dos objetos em movimento
        contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        for contour in contours:
            if cv2.contourArea(contour) < 1000:
                continue

            # Tirar uma foto
            timestamp = datetime.now().strftime('%Y%m%d%H%M%S')
            photo_file = f'photo_{timestamp}.jpg'
            cv2.imwrite(photo_file, frame)

            # Enviar a foto pelo Telegram
            with open(photo_file, 'rb') as photo:
                await send_photo_async(bot, chat_id, photo)

        # Exibir o resultado
        cv2.imshow('Frame', frame)

        # Pressione 'q' para sair
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Liberar os recursos
    cap.release()
    cv2.destroyAllWindows()

# Executar a função principal assíncrona
asyncio.run(main())
```

Figura 12 - Desenvolvimento (Código)

4.3. Funcionamento

Para executar o código é necessário saber a localização do ficheiro, que neste caso tem o nome **movimento.py** e que foi guardado no ambiente de trabalho.



Depois de descobrir a localização do ficheiro é necessário escrever esta linha de código na consola para abrir a câmera:

- **python3 /home/user/Desktop/movimento.py**

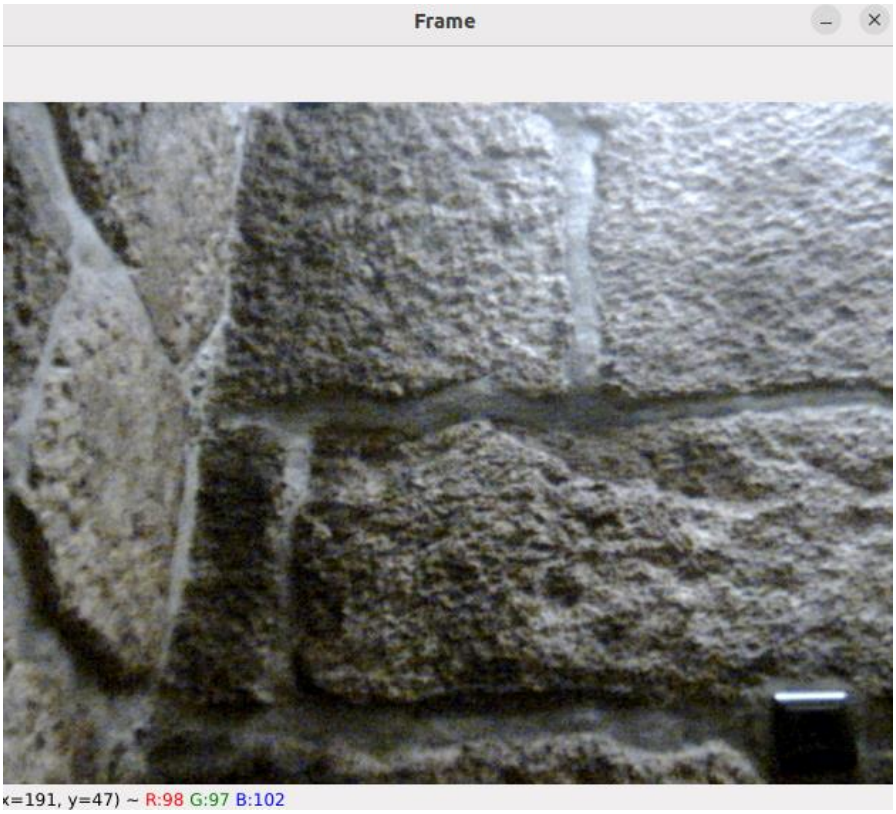


Figura 13 - Câmera

Se for detetado movimento (sendo este detetado através do cálculo da diferença entre os frames), as fotos são enviadas para o CameraMovimento_Bot (Camera).

Figura 14 – Notificações

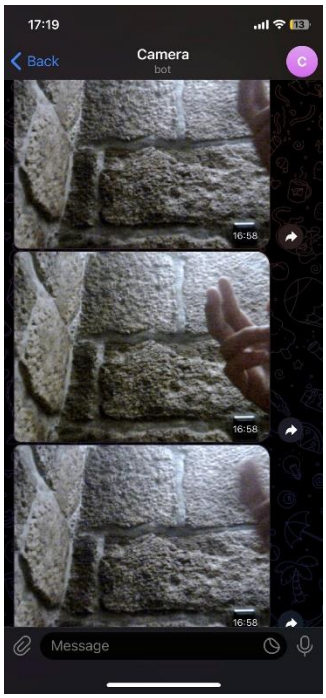


Figura 15 - Fotos recebidas

5. Conclusão

Neste trabalho foi possível aprender a utilizar diferentes periféricos nunca utilizados de forma conjunta pelo nosso grupo para a obtenção de um projeto que cumpriu o seu propósito. Inicialmente era para usarmos o Raspberry Pi OS em vez do Ubuntu, mas tivemos um problema com a ligação da câmera porque, quando era conectada ao Raspberry Pi, a barra do menu desaparecia, tornando difícil o uso do mesmo. Com o Ubuntu, conseguimos mais facilmente executar o que estava planeado. A montagem do sistema foi simples, logo tornou o projeto simples, mas eficaz.