Final Design

Target Acquisition & Image Processing

Prepared by Date 19/10/2020

Arik Intenam Mir, Image Processing

Checked by Date 19/10/2020

Name, UAVPayloadTAQ-G16

Approved by Date --/--/2020

Celine Blumer, Project Manager

Authorised for use by Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Felipe, Project Coordinator

Queensland University of Technology

CRCSS-EESE, GPO Box 2434

Gardens Point Campus

Brisbane, Australia, 4001.

This document is Copyright 2013 by the QUT. The content of this document, except that information which is in the public domain, is the proprietary property of the QUT and shall not be disclosed or reproduced in part or in whole other than for the purpose for which it has been prepared without the express permission of the QUT.

.

**Revision Record**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Issue/Revision Status | **Description of Change** | **Date** | **Approved** |
| 0.1 | Initial Issue | 19 October 2020 | Celine Blumer |

**Table of Contents**

[Definitions 3](#_Toc54018491)

[1 Introduction 4](#_Toc54018492)

[1.1 Scope 4](#_Toc54018493)

[1.2 Background 4](#_Toc54018494)

[2 Reference Documents 4](#_Toc54018495)

[2.1 QUT Sourced Avionics Documents 4](#_Toc54018496)

[2.2 Non QUT Avionics Documents 5](#_Toc54018497)

[3 Subsystem Introduction 5](#_Toc54018498)

[4 Subsystem Architecture 6](#_Toc54018499)

[4.1 Subsystem Interface 6](#_Toc54018500)

[5. Image Processing Subsystem Design 6](#_Toc54018501)

[5.1 Hardware components 6](#_Toc54018502)

[5.1.1 Raspberry Pi 3 Model B 7](#_Toc54018503)

[5.1.2 Raspberry Pi Camera 8](#_Toc54018504)

[5.2 Software Components 8](#_Toc54018505)

[5.3 Target Detection Design & Theory 8](#_Toc54018506)

[5.3.1 Type A Targets 9](#_Toc54018507)

[5.3.2 Type B Target 9](#_Toc54018508)

[5.4 Software Flow Diagram: 9](#_Toc54018509)

[6. Conclusion 10](#_Toc54018510)

[7. Appendix 11](#_Toc54018511)

[7.1 Appendix A 11](#_Toc54018512)

[7.2 Appendix B 12](#_Toc54018513)

[7.3 Appendix C 13](#_Toc54018514)

[7.4 Appendix D 14](#_Toc54018515)

**Table of Figures**

[Figure 1: Image processing elements 6](#_Toc54018516)

[Figure 2: Raspberry Pi 3 7](#_Toc54018517)

[Figure 3:Raspberry Pi Camera 8](#_Toc54018518)

[Figure 4:Software Flow Diagram for operations of image processing 10](#_Toc54018519)

[Figure 5: Type A targets 11](#_Toc54018520)

[Figure 6: Type B Target 12](#_Toc54018521)

[Figure 7: Raspberry Pi 3 pins layout 12](#_Toc54018522)

**Table of Tables**

[Table 1: Subsystem Interfaces 6](#_Toc54018523)

[Table 2: Raspberry Pi 3 Model B 7](#_Toc54018524)

[Table 3: Raspberry Pi Camera 8](#_Toc54018525)

# Definitions

|  |  |
| --- | --- |
| UAVPayloadTAQ | Unmanned Aerial Vehicle: PayloadTAQ |
| IP  UAV | Image Processing  Unmanned Aerial Vehicle |
| QUT  G16  REQ  HLO  SSH  OpenCV | Queensland University of Technology  Group 16  Requirement  High Level Objective  Secure Shell  Open Source Computer Vision Library |
|  |  |

# 1 Introduction

The Final Design, being the culminating step of the Systems Engineering process, outlines to clients and team members a full review of the subsystem at the time of the project’s completion. The document lists the requirements – sourced from the customer needs and system requirements – for the subsystem to fulfill, the design process undertaken over the course of the project and the features of the final subsystem that was constructed. The focus of this final design is the Image Processing subsystem. The connectivity of this subsystem in reference to the other subsystems is described in the Interface Control Document while the Systems Engineering approach towards this subsystem is explored in Group 16’s Project Management Plan.

## Scope

The aim of this report is to provide an overview of the subsystem throughout the entire project from the initial customer needs and system requirements, background research and preliminary designs to the final design that is presented to the client. To ensure that the solution has met the desired outcomes established by the client, the final design will address each HLO and REQ initially discussed at the commencement of the UAVP20G16 project.

## Background

Group 16 from EGH 455 was appointed by the QUT Airborne Sensing systems team to design and construct a UAV payload that can navigate under a GPS denied environment will be installed on an S500 UAV. The Payload will incorporate a board camera and other sensors to autonomously measure the telemetry, imagery and air quality data in a simulated underground mine environment, the sensory data gathered will be displayed onto a web interface in real-time. It is also a requirement for it to detect multiple markers placed around the mine using image processing techniques.

# Reference Documents

## QUT Sourced Avionics Documents

|  |  |  |
| --- | --- | --- |
| RD/1 | -SUP-CustomerNeeds | 2020: Customer needs |
| RD/2 | SR-UAVPayloadTAQ-20 | System Requirements Document |
| RD/3 | -PM-PMP-01 | Project Management Plan Document |
| RD/4 | -TAIP-TR-01 | Target Acquisition & Image Processing Subsystem Test Report |
| RD/5 | -TAIP-PD-01 | Target Acquisition & Image Processing Subsystem Preliminary Design Document |
| RD/6 | -PM-ICD-01 | Interface Control Document |

## 2.2 Non QUT Avionics Documents

|  |  |  |
| --- | --- | --- |
| RDn/1 | Raspberry Pi Camera | <https://www.raspberrypi.org/documentation/hardware/camera/> |
| RDn/2 | Raspberry Pi 3 Model | <https://www.raspberrypi.org/documentation/usage/gpio/> |
| RDn/3 | Cascade Classifer, OpenCV | <https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html> |
| RDn/4 | Aruco Tag  Detection, OpenCV | <https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html> |

# Subsystem Introduction

The Image processing subsystem will be responsible for capturing imagery data using the on-board camera as well as target identification. For achieving the results by the time this project reaches its completion stage, it is required to satisfy the relevant Higher-Level Objectives defined according to the customer needs (RD1).

* According to HLO-M-2, during flight, the on-board camera must be capable of searching the designated area for a target(s) (*Appendix A &B*). And once the target is located, the feedback from the camera will finalize the location of the target to the GCS. Also, to note that, the target detection must be done on-board in order to process imagery data using a Socket on Chip computer.
* All the tasks assigned for this subsystem shall be done according to the Systems Engineering Method, defined in HLO-M-5.

# Subsystem Architecture

The subsystem architecture and interfaces are discussed in this section. An overview of the subsystem architecture is shown below:

![A close up of a device

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDmRXhpZgAATU0AKgAAAAgABAE7AAIAAAAJAAAISodpAAQAAAABAAAIVJydAAEAAAASAAAQzOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEFyaWsgTWlyAAAABZADAAIAAAAUAAAQopAEAAIAAAAUAAAQtpKRAAIAAAADNTUAAJKSAAIAAAADNTUAAOocAAcAAAgMAAAIlgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADIwMjA6MDg6MjYgMTk6MTA6MzUAMjAyMDowODoyNiAxOToxMDozNQAAAEEAcgBpAGsAIABNAGkAcgAAAP/hCxtodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTA4LTI2VDE5OjEwOjM1LjU1MTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5BcmlrIE1pcjwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCAHHA9MDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD6RooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAryTR/iydGs9ZuPFrXuoD/hK7jR7BbK1RnRQoKJtXBbuM/MxJHWvW68lg+FetxXsUzXWnlU8Zvr5AkfP2cpgL9z/AFnt096cfi17f+3R/S43pB23/wCA/wBbFwfHrw0LYyz6T4itzBci3v0m0xl/s4lgFads7VBzwAS3HStvxj8S7LwXc7L3QPEV9bpF5019YacZLeBf9uQkDgcnGcVg678MtY1Pwv47063uLBZ/Eeox3Vq7u4VEXyuJCEyD8jdMjkVV8e/DXxh4r1nVooNR0S50TULREgj1WKWWTTpURhmBR8ilmIzJ94A9DtAM3fIn1/4C0+8dl37/AJ/5HTav8VNC0nXLXS0ttU1GSa1W8ml0+zaaO0gb7skpHKqeuQDjHOOKrz/F/RI/DmiaraaZrWoy62jyWmm2NoJroonDOUDYAHH8Xf648z8U69e/DHx1DFLqVrpS6joNtb3813ZTXETtCuz/AEVk5Z+TxIEUcE5yAJl+Dmp+K/hh4IuFtLD+0dNsnim07WjPFE6SMWBJhIdWHHHfdz0wX0dtr/8AyX+SMot8yUu342j/AJs9F1D4xeHrHTdEvIbPVtQ/trzltreytPMmWSIfPE0eQwfJ24GRkckDmrK/FLR10DW9UvLHUrFtEMQurK5hRbjMiKyBVDkZO8DBI5BFYmhfC270W+8DzQJpVtFoT3k17BaPPsLzx7R5Xml2OD1LMPUAdKx/FuiQ+If2gtJstJvlmgaKO58QWkOGCfZm32/mEdCxcDBxwAfSqsuZR7/h/S/Qt6R5uy+/W1vyPZYpPNhSQo0ZdQ2x8ZXI6HHen0UVIBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQByPxK8S6t4W8LQ3fh6Oykv7i+t7OIXyuYgZX25Owg9+35GsjSPFXjbR/GOm6L8RLLQjDrAdLK80V5dqSopYpIsvPKjIIwBjvni/wDFbR9Y1nwjbJ4d07+0r211K1u1tvPSHzFjkDEbnIA6f/WrFgsPGvjfxfo+peJvDVv4Xs9BaS5hjbUUu3up2Qqv+rACovJOeTkY74iLevr+Fl/wfmOW3y/G7/4HyN7TPi34G1nX5NF0zxBDcX8fmZjWKQK2wEttcrtbgE/KTkDjNZugfHHwZrmi6vqz35sbPSpgkjXEb7mRuEcALzuIbCjLYGSBnA5TSfDfxG1Pxn4V1fxdozebo97IL28/tdJFuFdGVZY7ddscSrwDgb2yCQcGprDSviX4a0PW9P8AD+gIMeIHvopft8AbULWSVi8abs+U2AvzNzgnGCBmlf8Ar1Wv3P8AB7ie+n9b/wCX4o9N8K+M/D/jbTpL7wvqKX9vFJ5UjCN0KtjOCrgEcHrityvNvhT4a17QtX8T3ev6W+nrqk1vcQK+pG+IHl4ZGlYl2ZTwSwx/dJFek1TVvwJjfqFFFFIoKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDk/Hviq78MWNq1hFE8tw7DdKCQoAHYEc8iuSj8e+NJY1ki0pXRwGVls5CGB6EHNaHxh/49NK/35P5LXbeGf+RT0j/rxh/9FivJkqtbFTpqbiklse3B0aGDhUlTUm29zzv/AITnxv8A9Aj/AMkpP8aP+E58b/8AQI/8kpP8a9XorX6pW/5/Mx+vUP8AnxE8o/4Tnxv/ANAj/wAkpP8AGj/hOfG//QI/8kpP8a9Xoo+qVv8An8w+vUP+fETyj/hOfG//AECP/JKT/Gj/AITnxv8A9Aj/AMkpP8a9Xoo+qVv+fzD69Q/58RPKP+E58b/9Aj/ySk/xo/4Tnxv/ANAj/wAkpP8AGvV6KPqlb/n8w+vUP+fETyj/AITnxv8A9Aj/AMkpP8aP+E58b/8AQI/8kpP8a9Xoo+qVv+fzD69Q/wCfETzvwd481bWPEo0rVreFNyv9xCjIyjOCCfYivRK8j8Nf8lnvP+vq7/m9euU8BOc6b53ezaJzKnCFWPIrJpMKKKK9A8wKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKa7rHGzyMqIoyzMcAD1oAdRXLah8R/C2nSGN9TWdx2t1Mg/76HH61TX4seGtw803kKn7rvbnDD1GM12RwOKkrqm/uOSWNw0XZ1F952tFZWk+J9F1zA0rUYJ3Iz5Yba+P904P6Vq1zThOm+Was/M6ITjNc0HdBRRRUFhRRRQAUUUUAebfGH/AI9NK/35P5LXbeGf+RT0j/rxh/8ARYrifjD/AMemlf78n8lrtvDP/Ip6R/14w/8AosV5tH/fanoj1q//ACL6XqzTooor0jyQorC8UeLbHwvaq1zma4k/1Vuhwze5PYe9cF/wtvV/NLjTbTyM4x8+fpuzj9K462NoUZcs3qd1DAYivHngtD1qiub8KeNbHxQjRoptryMZaB2zkeqnuK6SuinUhVjzQd0ctWlOlNwmrMKKKK0MwooooA8j8Nf8lnvP+vq7/m9euV5H4a/5LPef9fV3/N69crzcu/hy/wATPWzT+LD/AAr9Qooor0jyQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCrqd+ul6XcXskUsywIXMcK7mYD0FeOah4hn8WzRy6i8t2s7H7JoVhJ2B+9Mw6dPr3+UV7bXm/jrwm+n6dPe+G7URW80hfU4rb5ZZI+MhTzheCSoHfPIGK9jK6tGM+WS957P+tvVa9Lrc8jM6dWUOaL91br+t/R6dbPY5QXxsJBB/a1vpsp/5c9AtBLKD6NLkZI/32qR9UuYELza34ttkbnzLq13xkepBk4HsM0wzS2CpbSTTaKJRmLSdIj3XTqeQZZCQQSMHBJP+yBSsZtOxcSjxhpSnn7VK3mL9WXCflk9a+i5U+m/p/lr/AOTerPA5muv5/wCen/kvoRSJHdQG8mjtdTgi+dtR0dfs91bf7TxYGR7kf8CFdh4K8aXUmsR6DfXH9rrInmW2oQDLbfSVeqkdDnkH1yDXJfZdWvb2xk0iGG5v5pP9G1iw/dI6jO4TLjAYDrkA467q9b0Dw3Y6DHI9tbQx3VxhrmSJdqs2Odo/hXOSF6DNefmNWjClyVFdvby8/L8L7OK3O/L6daVXng7Jb+fl5/jbdPobFFFFfKn04UUUUAFFFFAHm3xh/wCPTSv9+T+S123hn/kU9I/68Yf/AEWK4n4w/wDHppX+/J/Ja7bwz/yKekf9eMP/AKLFebR/32p6I9av/wAi+l6s06KKK9I8k8b1SEeI/i61nek+T9oEO3OPkQdB9cH869djsrWKzFpHbxLbBdvkhBtx6YrzX4geHL/T9cHiXRlYrlXlMY5iccbseh/xz1pkXxfuVsts2lRvcgY8xZiEJ9duM/rXiUa1PC1aka+jbvfuj6CvQq4ulSlh9Ula19mUtUs08M/Fe0TSB5SSTRMIweFDnay/Trx717DXlHg3RNT8SeKh4k1lWWFHEyswwJHH3Qo/uj+gr1YkKpLEAAZJPaunL4vlnNKybukcuZSXNCm3eUVZvzForynxh43utav10Xwu0hjL7GlhJDTt6KR/D79/p1l8HeN7nS746H4oaRdr7EmmPzRN/dYnt79vpVf2hS9rydO/S5P9mVvY+0672627nqNFAIIyOQaK9A8s8j8Nf8lnvP8Ar6u/5vXrleR+Gv8Aks95/wBfV3/N69crzcu/hy/xM9bNP4sP8K/UKKKK9I8kKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAPOvFfgk2l/d69pF7c2VvNGzahDaqWlcdT5fpk9ecDk9OK4zQTDq+qLaeHjqmm6hKpeKY3YuI3xnImUKNoOMZ56jg17L4i8Q6f4X0OfVdXl8u3hHQcs7HoqjuT/nivMPgp4mj8QeJfE8hs4LNp2jnhhhXhEywK5+pU/UngV69LOFTp+yqay6f8Hv879vM4ZZBUrp4qlpTj8Xz/l7edrW38j0Twj4UtfCek/Z4G824lIa4nPHmN7DsBngVvUUV5lSpOrNzm7tnVTpwpQUIKyQUUUVmaBRRRQAUUUUAebfGH/j00r/AH5P5LXbeGf+RT0j/rxh/wDRYrifjD/x6aV/vyfyWuy0G4itfBemT3MixRR2ELO7HAUCMc15lLTG1fRHrVk3gKSXd/madxcRWtvJPcyLFFGpZ3Y4Cgd65DTPiZpOo64bBo3t4mbbDcyEbXPuP4c9v6VyHiPxHqHjrWE0jQ43+x7/AJE6ebj+N/QD0/rWxqXwqjTw5GNOlMmpxAs7McLMf7o9Pb9azniq9WTeGV4x38/JGsMHh6MEsU7Slt5ebPSqotomlNP5zaZZmUHPmG3Xd+eK4LwN45kt5l0LxGzRyIfLhml4KkcbHz+hr0skAEk4A6k13Ua1PEQ5l/wx51ehVws+R/f3EyEXnCqB+VeVeNPGc+u3f9g+HN8kUjeW7x9Zz/dH+z79/p1d418Zz63d/wBg+G90kcjeXJJF1mP91f8AZ9+/069R4K8FQ+G7UXF0Fl1KRfnfqIh/dX+p71x1ak8XN0aL91bv9Ed9GlDBwVeuryfwx/Vh4K8FQ+G7UXF0Fl1KRfnfqIh/dX+p70eNfBUPiS1NzaBYtSiX5H6CUf3W/oe1dZRXZ9Vpey9jbQ4Prdb23t7+9/WnoeW+CvGs+j3X9g+JN0axt5cckvBhP91vb0Pb6dPUgQRkcg1yXjbwVD4ktjc2gWLUo1+VuglH91v6GuZ8FeNZtIuv7B8Sbo0jby45JeDCf7re3v2+nTjpVZ4Wao1n7vR/ozvrUYYyDr0FaS+KP6oreGv+Sz3n/X1d/wA3r1yvIvDJB+M14RyDdXX83r12ry7+HL/E/wBCM0/iw/wr9Qooor0jyQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA8N+LHhjxVr1re6/rl3aado+mRlrWxEhkcknALYG3cxwOpxnH1xvhTpWreDfiTpB1a2aG212wYQS9VcMgkAz2YFQCDyM+leh/GSR7/AE3Q/DEDESa5qUcTY/55KQWP4Eofwr0CSxtZYYIpIEZLd1eIEf6tl+6R6Yrj9inVck9rH0v9pTp4CNKcVyz5lZaWWyfre+5YooorsPmgooooAKKKKACiiigDzb4w/wDHppX+/J/Ja5i51rVfFsGl+HdLiYQQQRxGMH/WMqgF2PZRjj8+tdP8Yf8Aj00r/fk/ktdF4E0Cy0jw3a3Num64vYEmllYcncoO0egGa8GpRnXxk4J2Wl/uPo6VeGHwNOo1eV3b1u9Sz4U8KWnhfT/LixLdSAefORyx9B6AVvUUV7cIRpxUYqyR4FSpOrNzm7tnGeOfA0evQtfacqx6ig5HQTAdj7+hrz6bxdr8uir4clZlcP5TM3EjDoIyT7/4V7pXGeOfA0evQtfacqx6ig5HQTAdj7+hrzMZhJO9Sjo3uu//AAT1sDjYK1LEapbN9P8AgE/grwVD4btRcXQWXUpF+d+oiH91f6nvXWV5n4H8cyW0y6F4jZo3Q+XDPLwVI42Pn9DXpldWDnSlSSpaJdP8zjx0K0azdbVvr0t5BRRRXYcIVyPjbwTF4jtzdWYWLUo1+VugmH91v6H/ACOuorKrShVg4TWhrRrTozU4OzR4n8O4pYPiFDDcKySxiVXVuqsFIIP417ZXkfhr/ks95/19Xf8AN69crgyxctKS/vP9D083lzVoy7xX6hRRRXqHjhRRRQAUUUUAFFFFABRRRQAUVja94q0nw55KajO7XVwcW9nbRNNPOf8AYjUFj7nGB3IrK/4Ti/bmPwJ4odT0by7Vc/g04I/EUAddRXJf8JvqP/Qg+KP++bP/AOSKP+E31H/oQfFH/fNn/wDJFAHW0VyX/Cb6j/0IPij/AL5s/wD5Io/4TfUf+hB8Uf8AfNn/APJFAHW0VyX/AAm+o/8AQg+KP++bP/5Io/4TfUf+hB8Uf982f/yRQB1tFcl/wm+o/wDQg+KP++bP/wCSKP8AhN9R/wChB8Uf982f/wAkUAdbRXJf8JvqP/Qg+KP++bP/AOSKP+E31H/oQfFH/fNn/wDJFAHW0VyX/Cb6j/0IPij/AL5s/wD5Io/4TfUf+hB8Uf8AfNn/APJFAHW0VyX/AAm+o/8AQg+KP++bP/5Io/4TfUf+hB8Uf982f/yRQB1tFcl/wm+o/wDQg+KP++bP/wCSKa3j9rYeZqvhLxLp9uPv3ElpHMqD1IgkkbA9cUAdfRVXTdTsdZ06G/0q7iu7Sdd0c0LhlYfUfy7VaoAKKKKACiiigAooooAKKKKACiiigAooqnqur6foenSX+r3cVnaxfellbA9h7k9gOTQBcorkF8fSXCiTTfB3ia+t25SdbSKAOPULNKjY/wCA07/hN9R/6EHxR/3zZ/8AyRQB1tFcl/wm+o/9CD4o/wC+bP8A+SKP+E31H/oQfFH/AHzZ/wDyRQB1tFcl/wAJvqP/AEIPij/vmz/+SKP+E31H/oQfFH/fNn/8kUAdbRXJf8JvqP8A0IPij/vmz/8Akij/AITfUf8AoQfFH/fNn/8AJFAHW0VyX/Cb6j/0IPij/vmz/wDkij/hN9R/6EHxR/3zZ/8AyRQB1tFcl/wm+o/9CD4o/wC+bP8A+SKP+E31H/oQfFH/AHzZ/wDyRQB1tFcl/wAJvqP/AEIPij/vmz/+SKP+E31H/oQfFH/fNn/8kUAdbRXJf8JvqP8A0IPij/vmz/8Akij/AITfUf8AoQfFH/fNn/8AJFAHW0VyP/Cc3y8y+BPFCIOreVavj8FnJP4A1saD4n0rxJHMdLuGM1u224tZo2int29HjcBl/EYPbNAGtRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRSMyohd2CqoySTgAVyR+IljdSOPDukax4hjjYq1xptsvkEjqFlkZEf/AICTQB11Fcl/wm+o/wDQg+KP++bP/wCSKP8AhN9R/wChB8Uf982f/wAkUAdbRXJf8JvqP/Qg+KP++bP/AOSKP+E31H/oQfFH/fNn/wDJFAHW0VyX/Cb6j/0IPij/AL5s/wD5Io/4TfUf+hB8Uf8AfNn/APJFAHW0VyX/AAm+o/8AQg+KP++bP/5Io/4TfUf+hB8Uf982f/yRQB1tFcl/wm+o/wDQg+KP++bP/wCSKP8AhN9R/wChB8Uf982f/wAkUAdbRXJf8JvqP/Qg+KP++bP/AOSKP+E31H/oQfFH/fNn/wDJFAHW0VyX/Cb6j/0IPij/AL5s/wD5Io/4TfUf+hB8Uf8AfNn/APJFAHW0VyX/AAm+o/8AQg+KP++bP/5Io/4TfUf+hB8Uf982f/yRQB1tFc7pPjfS9T1JdMuIrzSdTdS0dlqduYJJQOpQ8rJj/YY4roqACiiigAooooAKKKKACiikZgqlmIAAySe1AHm0v/FQftCwx/et/DunFyDyBNJ/9i4P/Aa9Krzb4QKdU/4STxVIDu1jUn8on/nkmdv/AKER/wABr0msaWsebuehmHu1VS/kSXz3f4thRRRWx54UUUUAFFFFABRRRQB5t8Yf+PTSv9+T+S123hn/AJFPSP8Arxh/9FiuJ+MP/HppX+/J/Ja63w5qNknhXSke8t1ZbKEEGVQQdg968yk0sbUv2R69ZN4Clbu/zZuUVV/tOw/5/bb/AL+r/jR/adh/z+23/f1f8a9Hmj3PK5Jdi1RVX+07D/n9tv8Av6v+NH9p2H/P7bf9/V/xo5o9w5JdjlvHPgaPXoWvtOVY9RQcjoJgOx9/Q1i+B/HMlpMuheI2aNkPlwzy8FD02Pn9DXof9p2H/P7bf9/V/wAa5rX/AAz4c8Qapb3txeQRSI377y5lHnLjgHnr059Pwrz61Fxqe2oNJ9V0Z6lDEKVL2GJTcej6o6+iqcV/psMSRRXdsqIoVVEq8AdB1p39p2H/AD+23/f1f8a9Dmj3PM5Jdi1RVX+07D/n9tv+/q/40f2nYf8AP7bf9/V/xo5o9xckux5b4a/5LPef9fV3/N69cryLwuyv8ZLt0YMrXN0QQcgj5q9drzst/hy/xP8AQ9TNf4sP8K/UKKKK9M8kKKKKACiiigAooooAKgvryLT9PuL25JWG2iaWQjsqgk/oKnrC8cf8k98Rf9gu5/8ARTUAZ3gDSWOljxPqsYbXNcjW5uJGOTDG3zR26+iIpAwOrZPU111Z3h7/AJFnS/8Arzi/9AFaNABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAcXdW6+EviDp91YKIdN8RzPa3sCjCLdiNpI5gOxZUdGx1OwnkV2lcl48/wCPrwn/ANjBB/6KlrraACiiigAooooAKKKKACiiigAooooAK4rTLWPxZ481HV9QXzbPQLn7DpkDcqs4QNNOR0LZfYp7BWx96u1rkvh9/wAe3iH/ALGC+/8ARlAHW0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXHfECxNjpp8YaVGE1bQozcFkHNzbLzNA394FQSM9GAI712NYnjT/AJELX/8AsGXP/opqANiCaO5t454W3RyKHRh3BGQafWZ4b/5FTSf+vKH/ANAFadABRRRQAUUUUAFFFFABRRRQAUUUUAcZ4ti/4SXxRpnhCQsNPlhfUNUVSR50KMqpCSP4Xdst6qhHeuxiijghSKCNY441CoiDCqBwAAOgrlE/5LRL/wBi+n/pQ9dbQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGX4h8P2PiXR5NP1FDhvnimQ4kt5B92RG6qynkEVS8D6vd6x4XjbVSp1Kzmlsb0qMBpoZDGzAdg23cP8Aeroa5L4f/wCq8R/9jBef+hCgDraKKKACiiigAooooAK5b4l6x/YXw41m7VtsjW5gjPfdJ8gI+m7P4V1NeafFf/ic634T8Jr8y6hqAnuFH/PKMc5/BmP/AAGs6rag7HbgaaniYKWy1fotX+COq8A6P/YPgHR9PK7XjtleQejv87f+PMa6GiiriuVWRzVKjqTc5bt3+8KKKKZmFFFFABRRRQAUUUUAc94u8JxeK7OCJ7prWSByySBN4wRyCMj0HeuS/wCFOf8AUd/8k/8A7OvTqK5KuDoVZc843fzO2jj8TRhyU5WXov8AI8x/4U5/1Hf/ACT/APs6P+FOf9R3/wAk/wD7OvTqKz/s7C/y/i/8zX+1MZ/P+C/yPMf+FOf9R3/yT/8As6P+FOf9R3/yT/8As69Ooo/s7C/y/i/8w/tTGfz/AIL/ACPLpvhDLFHvg1Rblh/yzaHy8/juP8qwJ/A09jMf7Va6tIv+eiW4lUfiGAr3CiolluHeyt95DzLGc3Mqn4K33WPIbD4dabqQH2TxRGzH+A2m1vyL1of8Kc/6jv8A5J//AGddvfeF9I1DLS2ixyH/AJaQ/Ic+vHB/GqH9ha3pnOjauZox0guxnj6//qqPqFGPxU7+jf8AmH9rY5by/Bf5HL/8Kc/6jv8A5J//AGdH/CnP+o7/AOSf/wBnXUf8JPf6dxr2kTQqOs8Hzp/n8a1rDXtM1LH2S7jZz/Ax2t+RpxwWCk7cuvq/8xrN8U9Of8F/kc34W+HUXhzWBqD6i126IVRRD5YBPBJ+Y54zXaUUV30qNOjHlpqyOWtXqV5c9R3YUUUVqYhRRRQAUUUUAFFFFABWF44/5J74i/7Bdz/6Kat2sLxx/wAk98Rf9gu5/wDRTUAXPD3/ACLOl/8AXnF/6AKp6j4oS11STTdN0u/1m9hRZLiGxEQ8hWztLPK6Jk4+6GLY5xjmrnh7/kWdL/684v8A0AVzVrrWmeEfFPiGPxNfW2lpqV2l7aXV5J5UU6eTHGUEjkKXUxnKg5wQcc0ddQ6HTaLrVpr2n/a7LzF2u0UsMybJIZFOGR17MD+B4IJBBOhXmfizxNYaxpFg+paVpsemXd7LHaXviGZkstqp8s7xkAPuO4xoxXcF3BlyDXNaNZWGpabp+lSfZbjT18ZMogtbZra3MZtHfCQszbYmyTtyVZWJ5DULV29PxaX6h/wfwTf6HuNFeOXPh7SLTTfFktrp1vC+ka/AumsiAfYVZbV2EP8AzyBZ2JC4Byc1V1qGG68R66Ne17QtH1kXzLp5vNLkuNSSLjyWtGE6sQeyxoRv3ghjuyLWSXdX++346h0frb8/8j22ivMhaWNreeP9du9MOp3tlNtjKbkmC/YodwjdQWi3Z5Kc8DrgVkeCn0mL4nWf9jyeGY7W50q686Lw5b7bdSrxFY5Jw2yd1Uk52IQCSVAYU0ru3lf8Lja0/ryPYpZBFC8jZIRSxx7Vx9t8RRNocGuT+FtctdHmhS4N/IbVkjhYAiRkSdpMAHJwpIHauii+w/8ACNr/AGR9n+wfZf8ARvsu3yvL2/Ls28bcYxjjFeTQ6Xqdj8HfDep3mt6nqfh1dPtm1fSDHbqDaGNd2xkhEhVOpUtllBG7PVpb36W/G+/3C6K3n+mx7QrBlDKcgjII70teMfEG80LWtQ1W2ltfDqyR2C/ZLjU42vbm9VotyCygBUquWI8yNj8ynK8ZEOuvpU0Ol6pqmqaDqFxJolmyafr++N5BtYmSzugSVlYkZ2I7ZVMlcrUprVv+t/8AINrLyv8Al/me21Rt9XgudcvdLjSQT2UcUkjMBtIk3bcHOf4Dngdq8uvpfCN34v16XxzbQ2882nWbW1vfnLxuYnysPrODgfJ8/TFQ2Q8OnUJx8UxanUzomn7hqWM+dsk3eXnjzs5+58/pSvo32/4P+RN7pPuey0V4bdJcz3NhB8RtQ0GwX+xLUwHxTYm4jkl+fzihM8arMMpu6vgrziuo0bRbW/8AF2gxazIdc+y6AJYpr62ZN7iZdkpjlywcDoWywyeck0/tcvm/wTf6Dk7K/p+LS/U9KoryTTvD9nafDW18T2FpGuq6devqMlyI900yRTSK8ZYfMR5LSKo5A4wK7XwGjXGgSa1MpE2t3D35ypBEbYEQIPpEsY+uaUXdX/r+t/uC+1uv9f5feQePP+Prwn/2MEH/AKKlrra5Lx5/x9eE/wDsYIP/AEVLV/xN410Dwjb+ZrmoRwuRlIF+aWT6IOfx6e9DaSuzSnTnUkoQV35G9RXJeBPGNz44t7vVItPNlpSSeTa+acyzEfeY44AGQMDPOeeK62lGSkroqtSnRm6c90FFFFUZBRRRQAUUUUAFFFFABXnXh/xCnh+x1QnT7zUJr7xVe2sFvaeWHZyXfrI6KBhD1Nei15B9mluxbQW97PYyP47uwtxbqheP93PyA6svtyppPdeq/MqKvf0f4JnoWl+J/t2sf2XqGj6ho940JnhjvjC3noCAxVopHHyllyCQfmGM81u15/4w8MCx8E+JdTv9X1DWL1dFuYIZr3yV8hChLBVijReSq5JBPyjnFVNX0rwzo+raHB4htLG28MvZzySfa0VbOS9Jiw85b5GkK79rPyTu71W/9ev6L7yXor/10/z+49LorxyyGkGPRf8AhJvJHggPqH2H+1T/AKL/AK1Ps27zPl2eX5vlbuNu3HajX00R9R0iCO80Oy8ICyk+wSeILOS6sXm805CB5Y0B2YMZyQU3eXhc5XRef4b79npsN6X8v87aHsdFeNapZaavhbQH1HxNoV7DEl0bNPEemyJpt0C4KBTI52FV+WNyzkxlioYc03WX0i90bQbvVH8P6dGli8Vto/igvNaTRrKF8yC6fBVygBDbWbaynaO49P69f8ifI9noryeS60C9v7a5+JFnbWOkSaJbnTrfWpDJFHLmTzgGlA3S7fKwSBIR2HNT+GdJj1bxJoH/AAkVo909voTTQxaipkZCLgeU7hxnzVTHzEbgS3qaPtKPr+Cb+/QG7K78vxt/meo1yMHjya8t7i6sPCWvXdnbzzQm5iNptYxOyOVQzhyMo2Btyewro9N/s/7Gf7H+y/Z/Nkz9l27PM3nzPu8bt+7d33ZzzXllho/iN/h/qN/4f1/UspqmoO+kxJbKk0QvJvMjjkMJdHYZwxY4J7dh6Jv/AIb5lf1+Z6tp9/bapptvf2EomtbqJZYZACNysMg4PPQ1YryPXdU8J6hY+HLW1tvDkWkSab5lhd+IMy20SAqvlRWxIWWYYAI3q6578g4tvHpk/gfw5Lquq6FGLWO9gi0/xNZutjKVuNoUF2/cSoq7VB3sqswCnBpvd+r/AF/yJT2TPdqovq8CeIYtHKSfaJbV7pWwNgVWVSM5znLjt615YbrwteeIdAm8aWVrZaadAdYoNVnM1spEyhCzygBsqMoXAJz0BqDT18PrqOir49EA0wafeiyGsj9yIPta+R5nmcZ8vZt389P4qUdbfP8ADm/yE3o2vL8bf5ntFFeJXa3DWOgw6xdabZeFi1/9nk8QWjXFoU84fZFkQyRgHyt2wuSMdBnFaenaJp1/J4U0+6vLXXtJkvb5o0SzaK0aMRErGkcjPvhVhlfmZeF28AYOl/6369mNu39f1dHrVYnjT/kQtf8A+wZc/wDopq4qHwbZa5pvjCS2tYF1ddUlWxvGjBe3aPypI1U9VTzFBKggE59afZXp8SeAfGHi2SJo/wC0tPmgt0dCrJDDCy4IPP8ArTKfxFKL5kn5J/eF/wA2vuO68N/8ippP/XlD/wCgCtOszw3/AMippP8A15Q/+gCtOmMKKKKACiiigAooooAKKKKACiuW8eeKL7wbpMOswWK31hFKEvYgxWRFbhXU9ODwQRzuHIqTwp4/8PeMof8AiT3o+0AZe0m+SVf+A9x7jIqOePNy31On6rWdH26jePft69iun/JaJf8AsX0/9KHqxr3jCfQbyOGTwxrF3FNPHbw3Ns9pslkfoAHnVhzxkqBx+NV0/wCS0S/9i+n/AKUPU3jf/VaD/wBhy0/9CNV1Xm0vvaRyydoyfZN/cjd027mvrFZ7rTrnTZGJBt7pomdcHqTG7rz14aorXV4LvWb/AEyNJBNYCIyswG0+YCRjnPbnIFeT+KYYbrxf4hTxRrGgaXKsif2UdV06Se6EPkph7N1nQhvMD8RqX3gdcqKteIjqMEPiTybpnK/2QL+7kt3/ANQP9dI8SsrbduS6hh8u7Pei+ife34/5dewO6kl6/wBf5HrtFeIXOm6W3hjWY9G13Rrywnn06OW38NWL2trE5u4xvDrNIglK8HaVbAQn+E1q6r4S8Pxax41sYtGso7O10SC8t7ZIQIoLhhcgzIg4WQiNBvADfKOaJNKLl2LUb/f/AJf5nrVVF1S0fWpNKWQm8jt1uXTacLGzMqnPTkq3Ht9K8f1ryLzXJD4v1bw9YQf2batp0niKye43qY/3jwOZ4wsgf720F/uEn7uHazp2n2EusT6jJb3OqXXgjEF/dWot7m7kSOYSvtf5w+zy9wPIGAelOXu3v0v+Cf8AkFNKdvO34tL9T2mivL9csrTwDpmj+KtDsEj8qB7a9jgiw1206Lsd8DLuZkiG48/OSTXc+H9Lk0DwlaWKj7Rc29vmQ5x50xG5zn/ack/jTlpfy/r+vUhO9vMp2njWyu9eTTks7xIpria0t79xH5E88QYyRqA+/I2PyUCnYcE8Z1Nb1eDQdGuNTvEkeG3ALrEAWOSBxkgd/WvDodfHh7VbbxJZanb6tq+ogvrGmXEbt/wj7nb50/kQgmJVA2PvwzEKS/XPq/xEdZPhzqjowZWiQhgeCN680npG/X9f6/rcJuzf4HU0V494t0TTrmH4navcWqSahYCOSzuWGXtHSziYPEesbZx8y4J2rnoK0tXttM8K6xr0mn6Sn2aXQY5ru2tpGtxcN5zAySOgJzgktJgtjOc0r6J9/wDK5clb+vRfqen0V4Vp7CHWNZXwZLoS3F14duHij8K2vlw+asi4VZQxSeVA5AYKhG4ZUbgKusfDa+JLM/CpbJ7seH9R2LpxAQzbYfL34483PXd8/Td2ok7K/r+F/wDIUFzS5f66f5ntFFeL+GbTTWura40rxFoJvFspzf2elaTJFdz5iIYXjmdyGD4O6VQ28EZyxBvJYaf4N+HvhXxnZWqxvpltDLqUkUWZbuGWFUkLkDLEN5b5OfuVT0v5W/G/+RKd2kut/wALfjqetVyXw/8A9V4j/wCxgvP/AEIVoeC9Mn0rwjZRXwAvpw11eYOf38rGST8AzED2ArP+H/8AqvEf/YwXn/oQoas7Ammro62iiikMKKKKACiiigArzPSWHiP9oDVb4HzLbw/YrZxsOglfOfx5kH4VteO/iNpvhCyeCB0vdakwltYRncxY9CwHIHt1PQUnww8J3HhfwuzaqS+ralKbu9YnJDt0XPt39yawk+eaiumrPUoweHw0609HJcsfO+79LafM7OiiitzywooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigA61k3/hjSdQJaa0VJD/y0i+Rv06/jWtRUyjGSs0JpPc5j+wda0znRtXMsY6QXYyPz/8A1Uf8JPf6cduvaRLEo6z2/wAyf5/Guno69az9lb4Hb8SOS3wszrHX9L1ED7LeRlj/AAMdrfka0ayb7wxpGoEtNZojn+OL5D+nX8azR4TvbJs6PrlxAg6Ryjev+H6Uc1Rbq/oF5rdHUUVzRh8Xwn5bmwuB/tKVP8hSi78XJw2n2Enurkfzaj2veL+4fP5M6Siuc+3eLP8AoE2h/wC2w/8AiqPt/iv/AKA9r/3+H/xVP2q7P7mHOuzOjqtfX8Gm24nu2KRbgrOBkLnoT7Z4/GsX+0PFX/QFt/8Av+v/AMVVbULvxFPp1xFeaJb+Q0Z8w/aF4GOv3u3WplV00T+5ic9NF+B1MM0VxCssEiyxsMhkbIP41i+OP+Se+Iv+wXc/+imrznStS1Gwul/syWQOx/1aDcH/AOA967bX7nUbn4X+IX1a0S2l/su5wFbO7903OO30yamjiFV0sKnUUzc8Pf8AIs6X/wBecX/oArRrO8Pf8izpf/XnF/6AK0a6TUKKKKACsCDx74QuryO0tfFeiTXMriOOGPUYWd3JwFChskk8Yreb7p+leVeGbDxP4h+D+m6Iun6Pb6deaekH25tQlkmSMj74h8gDeByB5mAcc8ULexVvdueoxXVvcSzRQTxyyW7hJkRwTGxAbDAdDgg4PYiqWr+I9D8P+V/b2s6fpnnZ8r7bdJD5mMZxuIzjI6etUPDeoXd3rXiO0upzLFp99HBbgqAVQ20TkZA5+ZmOTnrVbwfBBJqvia7mUNqMmpvDcM4+ZYkVfKT2XYQwHTLse5o62Xa/5f5iei1/r+rHUqyugdGDKwyCDkEUtcFfajpWm6NZaX4R1K/tI2uriKG10e0FzPK8bkPHGZg0cSK7YLMAijABUVk6Tr/ijUxpmmSardWU0mt3ljcXE1vbG58mOFpFBCBofMBwMqNvHI6ihat2/rb/ADRMnyrX+t/8j1OivMl1bxLZ2K6hceIprr7J4hj0kwNaQKlzC1ysW+XCbvMw5OUKL8q/L1yy28S+MdauJ9T0Ww1eaKHUZbaK2UWC2MkUUzRtvLyC4DkKxyMANj5CB8zSulJbP/JP9Ry0v/Xf/I9BtdIgtNZv9TjeQzX4iEqsRtXywQMcZ785Jq/XBR6vqSw+JtU1LxHLZWtjfSWNrElkkyRg7ArFFTzJJNz7VAbByMqx6t8H6zrUnja40zUZ9ZmsZdNW7g/tq3tYpsiTbuUQBSqsCPlkUMCOg5FTFqVrdVf8L/kJ2im+zt+NvzO/ooopjKGtaUut6VLp8t1cW0M42zG3KhpEP3kyynAYcEjDehB5q7HGkMSRRKERFCqqjAUDoBTqKAOI+J1rLfWvhy1truSylm1yJEuIgC0RMUvzDPcV4j42+E/izQrie/m8zW7ZiWe9hLPJ9XU5YfXke9e6/EKVID4YnlOI4ddikc4zhRFKSao6v4/lk3RaRF5S9POkGWP0HQfjmuPEqk1771PRwWbVMufu2ae6/wCDv+hwvwy8Nal4m8Gwy6T481PTRbu0UljCvywnOePmHBDA5x1JrsP+FZ+Jf+ik6z/3yf8A4upfCeoaBoi3N9cSsupXzbrp1jb5sH5c44J5PPXkiuj/AOE30P8A5+X/AO/Tf4VNP2XIuZ6+v/BKrZxOpUc6bST1s1F2+djl/wDhWfiX/opOs/8AfJ/+Lo/4Vn4l/wCik6z/AN8n/wCLrqP+E30P/n5f/v03+FH/AAm+h/8APy//AH6b/CtP3Hf8f+CY/wBq1v5l/wCAx/yOX/4Vn4l/6KTrP/fJ/wDi6P8AhWfiX/opOs/98n/4uuo/4TfQ/wDn5f8A79N/hR/wm+h/8/L/APfpv8KP3Hf8f+CH9q1v5l/4DH/I5f8A4Vn4l/6KTrP/AHyf/i6P+FZ+Jf8AopOs/wDfJ/8Ai66j/hN9D/5+X/79N/hR/wAJvof/AD8v/wB+m/wo/cd/x/4If2rW/mX/AIDH/I5f/hWfiX/opOs/98n/AOLpB8NfEjZx8S9YODg4U8f+P11P/Cb6H/z8v/36b/Cub8QeIbdtRi1Hw/eyRzkbJl2EBwOhIIwfT8qicqMVdO/zJlm1dK91/wCAx/yI/wDhWfiX/opOs/8AfJ/+LrS+FlrLZaDq1rcXL3csGtXcb3En3pWVwC59yRn8afpHj+KXbFq8XlN086IEqfqOo/DNSfDtg9nr7qcq2v3pB9vMrai6ctYGdTG1MTG0mtPJL8kjr6KKK3MArJ1XxX4d0G5S31zXtM02d03rFeXkcLMuSMgMQSMg8+1a1cJfT61D8W73+wbCwvWOiW3mC9vntgo8+fGCsUme/p+NLql/WzZSXut9v80v1Ottdc0m+tbW5sdUs7m3vHMdtLDcI6zsM5VCDhiNrcD+6fSpNR1Ow0exe91e+trG1jIDz3UyxRrk4GWYgDJOK4yGHWPDusaa91NbwTeIdeeS8tbT95FGn2R8IrsoJOYVYthSSTxjr1OrwWktzaSGK0uNVt/Ml06G5uDEGk2bSRgMejYLBWIDHjmqe1xf1+L/AMi/a3VvfWkV1ZTx3FvMgeKaJw6Op5BBHBB9RSW13bXiyNaXEU6xyNE5icNsdThlOOhB4I7VzXw/d10vU7K6VI7601ScXcMI/cxSSETYjPVl2yqckAkkkheg5jw9qf8AwjP9t6jsLxXmoamojHWS5iuJDGo/2nUsP+ACk9/lf8v8wt7t/O35/wCR6ZBe2t1JPHa3MMz20nlzrHIGMT4B2sB0OCDg9iKmryjR21DQJtW0ixu/Kvb/AF+3s5r/AMpWMbNZRvJKFPy7mKtjII3MMgjg6V5reu6ZrM3hYaxLczT3VpHDq0ttF5tvHMszMGCqsbOPIYKdmP3i5DY5dv0+92/zBq33X+Wv+R6AlzBJcyW8c8bTxBWkiVwWQNnaSOozg49cGi5uoLO3e4vJ44IUGXklcKq/UngV5Ze6pq3hDV/FLrey6xesdLtbeaSGITfvpHQbgPLjLDecfcU/LnuS/Wp/E/8AwiOv2+tWuqnTvssbw3Or/YhOJfNUFB9lYqVxgglQQQeTkYXRtf11IcrJs77UIdL0vVH8TapfJZLDafZXluJljhRC4bJLYwd2B171JpHiXQvEDSjQda0/UzDgyiyukm8vOcZ2k4zg9fSsX4nGVfAkxt0SSYXtkY0kcorN9riwCwBwM98H6GqHm6x/ws7QrnxJpNlZeZaXVray6ffNchnOyQrJvijIG2IkYB5znHFEe3r/AJ/mW0kubv8A8BHeUUUUCIL23ku7GWCG7ms3kXatxAELx+671Zc/UGsHxFpttpPwt1fTrFDHb2+kXEcYJycCJuSe57k9zXS1i+MxnwHr4H/QNuf/AEU1A02ndHCaN8I4LvQbC5/4SvxHEZraOQpHeAKuVBwBt6VcPwfiT/mc/Eq5OBm9HX8qtQeN49P8O6fZ2EBluIrSKN3k4RWCAEY6nn6VX0TxDFJrDah4hvXZolxBGEJVSepAHA44/GuGU6Clyo6Xm+JvZT/Bf5Cf8Kfx93xr4mH/AG+//Wo/4VDKPu+OPEw/7fDXUf8ACb6H/wA/L/8Afpv8KP8AhN9D/wCfl/8Av03+Faf7P3X3l/2riP5/wX+Ry/8AwqS5H3fHniYf9vho/wCFTXw6eP8AxMP+3tv8a6j/AITfQ/8An5f/AL9N/hR/wm+h/wDPy/8A36b/AAo/cd194f2rX/nX3L/I5f8A4VRqQ6fELxMP+3tv/iqP+FVasOnxF8S/+BTf/FV1H/Cb6H/z8v8A9+m/wo/4TfQ/+fl/+/Tf4UfuO/4h/atf+dfdH/I5f/hVmsjp8RvEf43DH/2aj/hV2ujp8R/EH4ysf/Zq6j/hN9D/AOfl/wDv03+FH/Cb6H/z8v8A9+m/wo/cd/xD+1a/8y+6P+R5f8Q/Cl/4X8GXd3qfj7WL1JAIY7OZyVuGb+E5Y8YyT7CuC8G/CzxX4knhu7WF9LtQQy3txmP8UH3m9iOPcV7l4n1jQdasVeK4f7ZbfNbsUcAHIJGOnYc9qh0jx9cwbYtWj+0R9PNQAOPqOh/SuWaoupq9DupcRVKFJ00k5Prpb7kl+IvhzS7rRvicLK/1a51aePw8m66uQoZv9Ibjjt9ST716BXE6df2+pfF17mzcvE3h9MEqR/y8P6121enG1tDxpTc25PqFFFFMkKKKKACiiigDIv8AT7LxDdQo9+0sWm3IaezheNkeVdroJflLAqdrgAr1GcjFaUV1bzzTQwzxyS27BZkRwWjJAYBh2JBB57EVzPg3/kYPGf8A2Gx/6R21ZGr+IddGsanZWOoC3Eev2NlExgRvLhlhjaQcjkksxBPQ+3FEtOVd7fe7f5j6Sb6X/C53s9zBaqjXM8cKu6xoZHC7mY4VRnqSeAO9S15T4k1PVdNjvdPup7vxA+m6zpctrujhjnl8xwfLOxUT7ynBwMA89M12PgXUb3WvD41fUb/z57t2L2axqqae6kq0AwNxKkFWLE5IJG0ECiOqb7P9F/n/AFdEX1t/XX/I6WiiigohvbVb6xntZJJolmjKF4JWjkXIxlWUgqfQisnSvDP9n6kNQv8AV9Q1i7jhMEMt95I8lGILBRFGg+YquSQT8o5rcooAKztZ0WLXLeG2u550tVlWSWCPbtuApyEfKk7cgEhSM4wSQSDo0UAFeQ6L4M1nXdS8RXeneMtR0eH+3LtPs1upK5D/AHvvDnn9K9erkvh//qvEf/YwXn/oQqZRUtzejXqUG3D8k/zTMT/hWfib/opWs/8AfJ/+Lo/4Vn4m/wCilaz/AN8n/wCLr0iio9jD+mzp/tHE91/4DH/I83/4Vn4m/wCilaz/AN8n/wCLo/4Vn4m/6KVrP/fJ/wDi69Ioo9jD+mw/tHE91/4DH/I83/4Vn4m/6KVrP/fJ/wDi6in+E2r3y+XqfxB1y5h7xhioP/jxH6V6bRS9jD+mwWZYlapr/wABj/kcd4T+F3hrwhcC7sreS6vh0u7tg7r/ALvAC/UDPvXY0UVpGMYq0UclatUry56sm35hRRRVGQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVPVdPOqWLWhmaGOQjzCg+Zl9B6Zq5RSaTVmJq6sUdN0aw0mPbY26xkjBc8s31Jqh44/5J74i/7Bdz/6Kat2sTxspfwB4gVRlm0y5AA7nymoSSVkCSWiLXh7/kWdL/684v8A0AVo1m+HGV/C2lMhBVrKEgjuNgrSpjCiiigA61V0zTbTR9Lt9O02LybS2jEcUe4ttUdBkkk/jVqigCraaZaWN1eXNrFslvpRNcNuJ3uEVAcE8fKqjjHSs7VPCOj6vfm9uY7qC6ZBHJNY309o8qjoHMLqXA5xuzjJxjJrboo3C5iXfg/RLuysrX7I9rHp4K2psLiS0eFSMFVeJlYKcDIzg4GegpNN8G6Do7wtptgIDDcPdRgSuQsrx+WzYLYyV6+pJPUk1uUUA9dzLk8NaTLavbvaZie9W/ZfMfmdZBIHzn+8oOOnHTFVZ/BOg3GpvfS2kxeWYTywLdzLbyyDBDvAH8pmyAclScgHqK3qKFpt/X9WQGXN4a0m40+/sZrMPb6hMZ7lC7fNIcfODnKkFVIKkYIBGDUGleD9F0bVDqVjBOb5oTA91cXk1xI6Eg7WaR2LAFRjOdvOMZOduihabf10DfQKKKKACiiigDkfHnNz4Uz/ANDBB/6Klq7q/gzTdS3SQL9jnP8AFGPlJ91/wxVLx3zeeElHVvEEOB64hmJ/QE/hXXVEoRmrSQpRUlZmF4b067021ew1GKORYmzDMvIZT29QQfX19q2vIi/55J/3yKfRTjFRVkCSSsM8iL/nkn/fIo8iL/nkn/fIp9FVYYzyIv8Ankn/AHyKPIi/55J/3yKfRRYBnkRf88k/75FHkRf88k/75FPoosAzyIv+eSf98iuZ1/QbzXtSjgiCWtjAMtIRy7HrgDrgeuO9dTRUTgpqzJlFSVmZGk+GNN0jDQw+bOP+W0vLfh2H4Vk/D7/j28Q/9jBff+jK62uR+H3EHiJT1XxBe5Hpl8j9CD+NOMVFWihpJKyOuoooqhhVVdMtE1iTVFixeSwJbvLuPMaszKMZxwXY5xnmrVFAFW80y0v7iznu4vMksZvPt23EbH2MmeDz8rsMHI5qDWdB07XoIo9ShkYwSeZDLDO8EsTYIykkZV1JBIOCMgkHg1o0UAU9K0my0WxFppsPlRBmdtzs7uzHLMzsSzMTyWYknuapv4U0SSCKGXT0kji1A6nGHZm23Jcv5gyeu5icdOelbFFHW4dLGTdeF9GvIb+O5sg66hMtxcHzGDGVVVVdWByjAIuCuCCMjnmoIvBmhRaTdaf9jeWK7kEs8k9zLLO7rja5mZjJuXA2tuyuBgjFbtFAHP2vgbw9aWuoW6WDTJqaqt61zcSzvcbc7SzyMWJGeGzkYHPyjCxeCNCitbuB7e4uReIsc0l5fT3ErIp3BBJI7OFB52ggZJPet+ijzFZWsUdY0ey17S5NO1SN5LaVkZljleJgVYMpDIQwIZQcgjpVTS/Cul6Re/bLYXk9yEKJLfahcXbRqcZCGZ22ZwM7cZwM5wK2aKB+QUUUUAFYnjT/AJELX/8AsGXP/opq26w/GzBPh/4hZyAq6Xckk9h5TUAVLDw7p2seE9KN1AFl+ww4mj+Vx8g79/xqLRfDt54f1hgNl3YXA2s2MNGexI/Tj1rb8OqU8L6UrDDLZwgg9jsFaNZOlBy5upDhFu4zyIv+eSf98ijyIv8Ankn/AHyKfRWtixnkRf8APJP++RR5EX/PJP8AvkU+iiwDPIi/55J/3yKPIi/55J/3yKfRRYBnkRf88k/75FHkRf8APJP++RT6KLAYviKyurzT/sWmQxq9wdskrYAjTv8An0496qaR4I0+w2yXn+mTDn5xhB9F7/jXS0Vk6UHLmZDhFu7ORjAX4zyhQAB4fjAA7f6Q9ddXIrx8aJM8bvD67ffFw2f5j8666tSwooooAKKKKACiiigDnrnwNotzqV3f7tUt57yQS3H2PWLu2WRwoXcUjlVc7VUZx2q43hnSGuZLhrUtLJdRXjuZnJaaJQiN17KoGOhxzmtWigDMuPDmlXV5Ldz2u6eWaCZ38xhl4TmM4Bxwe3fvmpbHRrDTb6/u7GAwzahKJrnEjbXcKF3bSdoJAGSAM4Gc1eooWm39f1YAooooAKKKKACiiigArkvh/wD6rxH/ANjBef8AoQrra5L4fcweImHQ+IL3B9cPg/qCKAOtooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmyRpLE0cqh0cFWUjIIPUU6igDgfD+sp4D8rwp4qlFraQHytH1SZsQ3EH8ETueFlQYXBxuABGckDvI5EljDxOrowyGU5B/GmXNtBeW7293DHPDIMPFKgZWHoQeDXMSfCzwHJIXbwho2WOTiyQD8gKAOsorkv8AhVPgL/oUNH/8BE/wo/4VT4C/6FDR/wDwET/CgDraK5L/AIVT4C/6FDR//ARP8KP+FU+Av+hQ0f8A8BE/woA62iuS/wCFU+Av+hQ0f/wET/Cj/hVPgL/oUNH/APARP8KAOtorkv8AhVPgL/oUNH/8BE/wo/4VT4C/6FDR/wDwET/CgDraK5L/AIVT4C/6FDR//ARP8KP+FU+Av+hQ0f8A8BE/woA62iuS/wCFU+Av+hQ0f/wET/Cj/hVPgL/oUNH/APARP8KAOtorkv8AhVPgL/oUNH/8BE/wo/4VT4C/6FDR/wDwET/CgDrap6nq2naNZvd6vfW9lboMtLcShFH4mue/4VT4C/6FDR//AAET/CrWn/DzwdpV0tzp3hfSbedTlZUs03KfY4yPwoAzNKefxp4utvELW81voekq40zz0KPeTOu1rjaRlUCEqucE72PTFdtRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXD6hNL4E8VXusyQSS+HNYZJL54ULtYXKqE84qOTG6qgYgfKVBPBJHcUUAVdP1Ow1a0S60q9t723cZWW3lWRSPqDVquavvhz4M1O6a4vvCukTTMctI1mm5j6k45/Gq//CqfAX/QoaP/AOAif4UAdbRXJf8ACqfAX/QoaP8A+Aif4Uf8Kp8Bf9Cho/8A4CJ/hQB1tFcl/wAKp8Bf9Cho/wD4CJ/hR/wqnwF/0KGj/wDgIn+FAHW0VyX/AAqnwF/0KGj/APgIn+FH/CqfAX/QoaP/AOAif4UAdbRXJf8ACqfAX/QoaP8A+Aif4Uf8Kp8Bf9Cho/8A4CJ/hQB1tFcl/wAKp8Bf9Cho/wD4CJ/hR/wqnwF/0KGj/wDgIn+FAHW0VyX/AAqnwF/0KGj/APgIn+FH/CqfAX/QoaP/AOAif4UAdbRXJf8ACqfAX/QoaP8A+Aif4Uf8Kp8Bf9Cho/8A4CJ/hQB1UsscETSTSLHGoyzO2APxrhNc1WL4hSf8Ix4bf7XpTyAaxqcRzCsSnLW8b9Hd8bTjIVSc84Facfwt8CRSK6eENG3KcjNkh/QiungghtYEgtokhijGEjjUKqj0AHSgCQAAAAYA6AUUUUAFFFFABRRRQAUUUUAFFFFABRRRQByvi/TtQhvtO8TaBbG71DShJHLZqwVru1kx5kanpvBRHXPGVx3rS0HxXoviW38zSL6OWReJbZzsmhburxn5lI9CK2KxtZ8H+G/EUgl13QdO1CUDAkubVHcD03EZoA2aK5L/AIVT4C/6FDR//ARP8KP+FU+Av+hQ0f8A8BE/woA62iuS/wCFU+Av+hQ0f/wET/Cj/hVPgL/oUNH/APARP8KAOtorkv8AhVPgL/oUNH/8BE/wo/4VT4C/6FDR/wDwET/CgDraK5L/AIVT4C/6FDR//ARP8KP+FU+Av+hQ0f8A8BE/woA62iuS/wCFU+Av+hQ0f/wET/Cj/hVPgL/oUNH/APARP8KAOtorkv8AhVPgL/oUNH/8BE/wo/4VT4C/6FDR/wDwET/CgDraK5L/AIVT4C/6FDR//ARP8KP+FU+Av+hQ0f8A8BE/woA62iuS/wCFU+Av+hQ0f/wET/Cj/hVPgL/oUNH/APARP8KAH+IfG9vaSNpPhvy9Y8QyjbDZQPuWEnjzJ2HEcY6knk9ACa0/C2hL4a8NWmmecbiWMM887DBmmdi8jn/edmP41a0vRtM0O1+zaNp1rp8Gc+VawrEpPrhQKu0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFcX4j+Iv8Awj2ty6d/wjWt6h5YU+fZ22+NsgHAPtmplJRV2bUaNSvLlpq7O0orzf8A4XB/1JXib/wD/wDr0f8AC4P+pK8Tf+Af/wBeo9tT7nV/ZuK/l/Ff5npFFeb/APC4P+pK8Tf+Af8A9ej/AIXB/wBSV4m/8A//AK9HtqfcP7NxX8v4r/M9Iorzf/hcH/UleJv/AAD/APr0f8Lg/wCpK8Tf+Af/ANej21PuH9m4r+X8V/mekUV5v/wuD/qSvE3/AIB//Xo/4XB/1JXib/wD/wDr0e2p9w/s3Ffy/iv8z0iivN/+Fwf9SV4m/wDAP/69H/C4P+pK8Tf+Af8A9ej21PuH9m4r+X8V/mekUVn6Dq39uaHbaj9juLLzwT9nuU2yJhiPmHbpmtCtE7q6OGUXCTi90FFFfK9x4ps/s+uSH4j+L4vGEOpXKado9tcTS28jLMRDHs2FSDwNu7HbHai+thW0ufVFFeSzjxB8QPG//CN3uvan4dtdI0q2ur5dIlEE891MDx5mDhFwfl7n8CMfVrrxRoOkeNfC0PiHUL+bQLW31jTNQlmP2kwht7wyuuC/3GHPUH0wA9Fu+/4b/k36CXvbeX42t+aPcqK8JvPG2r6j8W7DxFpep3Q8LWc1hp1zbJKfIla6iZ/MKg4JUyRDOD1HIxWPr2va1f6Npeqyar4q+ya54kvTFBoFy/2o2qIUjSJc4xlN2OnJNKV47/1ql+twVnt/Wjf6WPo6ivBYrq6074XeNr/T7v4kWlzBYoY5PFkpRkOT80BHIP8AePutZnh/XrKbxR4cX4c+PvF3inUpL2IalYalNNLbx2hGJnbfGoGMgA5PJGOapaz5PT8b/wCWonpG/r+B9G0UUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABWB4O8L/8IlpFzY/bPtfn31xeb/K2bfNkL7cZPTOM9/QVv0UdbgcT4s8AXmseIoPEfhbxJceGtbSH7NLcR263EdxDkkK8TEAkE5B7enQifwp4Aj0G21aXWNTm13Vtb/5CN/PGsfmqFKqixjhFCk8c9T2wB19FKys0HW55bpPwQsdM+EGqeBm1QznUZmna/Ntgo+VKHZuOcbF7jPPTNXdT+Fl2/hnwppvhvxH/AGNdeGuYbz7As/mMYyjHYzADOSec9fxr0Wim9d/L8Ng/4P47nnUvw+8W6r4a1rRvFPj/APteDUrQwRH+xooPs7ZB3/IwLcDGCRWnr/w+GpazoWuaPqQ0rWtIKxteLb+YLq3xhoZF3LlT1HPy5OOTmuyoo2d/T8L/AObF0t/X9aBRRRQMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACisGfx14Rtb6SyuvFOiw3UchjeCTUYldHBwVKlsg54xW1PcQ2ttJcXU0cMESF5JZGCqigZLEngADnNHS4dbElFZ8viDRoNFXWJ9WsY9LYBlvnuUEBBOARITt5PA5qnYeNvCuq30dlpfibR726lyI4Le/ikd8DJwqsSeATR5AblFVbLVNP1JrhdOvra7a1lMM4gmVzFIOqNg/Kw9DzVqgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArF1G18QyXztpt/bQ25xtSRMkcc9vWtqiplHmVhNXOc+xeLf+gpZ/8Afr/7Gj7F4t/6Cln/AN+v/sa6Oio9ku7+8nkXdnOfYvFv/QUs/wDv1/8AY0fYvFv/AEFLP/v1/wDY10dFHsl3f3hyLuznPsXi3/oKWf8A36/+xo+xeLf+gpZ/9+v/ALGujoo9ku7+8ORd2c59i8W/9BSz/wC/X/2NH2Lxb/0FLP8A79f/AGNdHRR7Jd394ci7s5z7F4t/6Cln/wB+v/saPsXi3/oKWf8A36/+xro6KPZLu/vDkXdlewS7jsY11CVJbkZ3ugwDycdh2xViiitErKxYUUUUwPHPBPhLwtry/EC48TaLpt3t8R30b3V1boZIo9qk4kI3KBknIIx1rm28TXqfs6+FdEls9QvrjWpzZC3sojJcSWEUjFyidW/dKq9hhs5xXpWofBL4e6rrVxq2oeHlnvbmdp5pGvJ8O5OSSofbye2MV06eF9Fj1ix1SKwjjutOtmtLMxkqkERxlVjB2joBnGcDHSmn7qX+H8F+v5XLlK8m13b+/wDq/qkeb/Cq603WbXX/AAZq3h+6t9Ptbn7ZYaVr1gEcWsjbgDG+4FUkDAHnjbWR4Vj8P+DvD3xC8TJoWnfbdH1+9isHW1QSR/KixxRsBlQS2MD1NexyeH9Ml8Sw+IHtf+JpDbNapcB2B8oncVIBwRnnkHHas6TwD4algnhk03MVxqf9rSp9olw91kHzCN3IyB8v3eOlLpbyt+K1+5feJWSt53/B6fe/uPIvhRqzeHPHOm6fPofiDSYtcsBDezaxYNAtzqKbpPMRixyXUuDnB+Va+gKzdZ8P6Z4gjtE1e1+0Czuo7u3O9kMcqHKsCpB49Oh71pVTd1/X9f8AAsRaz/r+v+DcKKKKkYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH//Z)

Figure 1: Image processing elements

## Subsystem Interface

|  |  |
| --- | --- |
| **Interface** | **Description** |
| 1. Raspberry Pi to Camera | The Raspberry Pi Camera will be connected to the on-board camera through the CSI camera socket. Which will also power the camera as well. The data stream will bi-directional. |
| 1. Raspberry Pi to Imagery Link | The on-board Raspberry Pi computer will be setup with Wi-Fi and SSH for transmitting the imagery and telemetry data to GCS. |

Table 1: Subsystem Interfaces

# Image Processing Subsystem Design

## 5.1 Hardware components

Based on the QUT Sourced Avionics Documents, it was decided to implement the image processing subsystem using Raspberry Pi 3 Model B micro-computer and for camera, the Raspberry Pi Camera. For this project, the Raspberry Pi 3 was chosen as the on-board computer. The Raspberry Pi camera will be connected to the micro-computer via the CSI port. The CSI port also acts as the data link between the camera and the computer.

### 5.1.1 Raspberry Pi 3 Model B

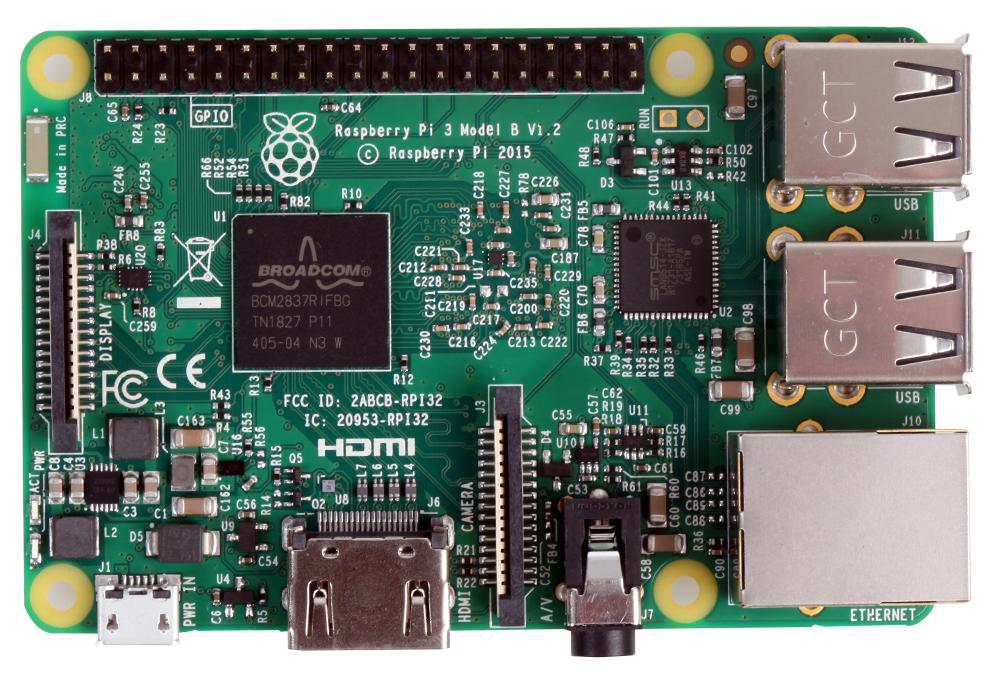


Figure 2: Raspberry Pi 3

For this project, the Raspberry Pi 3 Model B microcomputer is an excellent choice as an on-board computer as it is very light in weight. It is pretty fast in processing speed and comes in a reasonable price as well, thus remaining under the budget limitation. The GPIO pins layout diagram is given in the Appendix B.

The specifications of the Raspberry Pi 3 Model B are:

|  |  |
| --- | --- |
| **CPU** | Quad Core 1.2GHz Broadcom BCM2837 64bit CPU |
| **GPU** | 400MHz VideoCore IV |
| **SoC** | BCM2837 |
| **Instruction Set** | ARMv8-A |
| **RAM** | 1GB SDRAM |
| **Storage** | Micro-SD |
| **USB** | 4 USB 2 ports |
| **Network** | 10/100 MBPS Ethernet, 802.11n Wireless LAN, Bluetooth 4.0 |
| **GPIO Pins** | 40 |

Table 2: Raspberry Pi 3 Model B

### 5.1.2 Raspberry Pi Camera

A circuit board

Description automatically generated

Figure 3:Raspberry Pi Camera

The specifications for the Raspberry Pi camera are given below:

|  |  |
| --- | --- |
| **Sensor Resolution** | 3280 × 2464 pixels |
| **Sensor Image area** | 3.68 x 2.76 mm (4.6 mm diagonal) |
| **Focal length** | 3.04mm |
| **Horizontal field of view** | 62.2 degrees |
| **Vertical field of view** | 48.8 degrees |
| **Video modes** | 1080p30, 720p60 and 640 × 480p60/90 |
| **Network** | 10/100 MBPS Ethernet, 802.11n Wireless LAN, Bluetooth 4.0 |

Table 3: Raspberry Pi Camera

## 5.2 Software Components

For the image processing subsystem, the software components are very crucial as it requires capturing imagery data and process them to find two different types of targets. Namely, type A, set of normal targets which must be identified using AI algorithms, requiring training for target detection and type B, an ARUCO tag, which will require open source ARUCO library algorithms for target detection. In order to successfully process image capturing followed by processing them on-board to detect targets, it was decided to make use of Haar-Cascade Classifier for type A targets (RDn/3) and OpenCV’s ARUCO library (RDn/4) for detecting the type B target.

## 5.3 Target Acquisition Design & Theory

Target detection is an important aspect of this subsystem. As per the Customer Needs (RD/1) and System Requirements (RD/2), the goal is to be able to be detect two types of targets (*APPENDIX A*). The implementation of target detection mechanism is tested, finalized and verified for final design of the subsystem.

### 5.3.1 Type A Targets

After background research and discussion, it was proposed to make use of the Haar Cascade classifier for detection of type A targets, which makes use of an open source computer vision machine learning software library, called OpenCV. Haar Cascade classifier operates by training a cascade function, which takes various positive and negative images as its input, for detecting targets from images.,

Haar Cascade classifier is superior than its contemporaries, because it has less chances of detecting false positive results. This cascade classifier will be implemented using Python, as there are available libraries for supporting the task at hand. An example of the Haar Cascade Classifier used for face recognition is attached in *Appendix C.*

Though it was intended to detect and process the type A targets using the Haar-Cascade classifier and discussed in the preliminary design report (RD/5), unfortunately due to time constraints and other limitations, it was not possible to implement the mechanism for type A targets.

### 5.3.2 Type B Target

For this project, the type B target will be an ARUCO tag, used extensively in the field of augmented reality applications. Generally defining, it is a synthetic square marker composing of black borders, containing a binary matrix holding the id of the tag. The outer black borders are beneficial for detection while the binary matrix allows for identifying the tag as well as correction and error detection techniques. The Python script used for detecting and processing the ARUCO markers or type B target is attached in *Appendix D.*

## 5.4 Software Flow Diagram:

The Software flow diagram seen in figure 4, outlines how the imagery is captured, processed and transferred throughout the system.

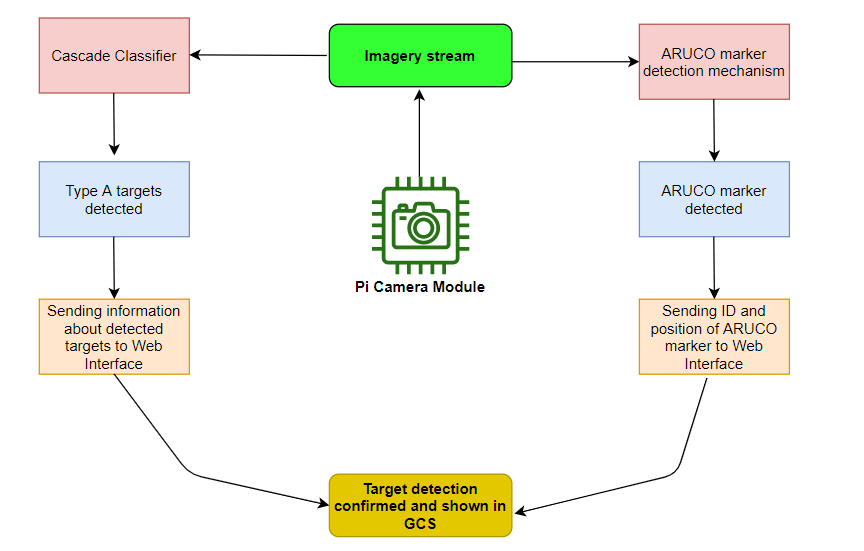


Figure 4:Software Flow Diagram for operations of image processing

# Conclusion

The Image Processing subsystem works by the unison of the discussed hardware and software components in order to satisfy the customer needs (RD/1), by completing the system requirements(RD/2), REQ-M-01, by being less than the weight limit of 250g, REQ-M-03, by being able to communicate with the GCS to transmit video data and target detection, REQ-M-04, partially, as the target identification system is only able to detect the type B target(*APPENDIX A, fig-6*) and is able to alert the GCS of its type, REQ-M-08 and REQ-M-10 as the preliminary design was completed by week 7, by maintaining the integrity of systems engineering approach. Finally, the subsystem also satisfied REQ-M-13 and REQ-M-14, as the processing was done on-board while the UAV was moving at maximum speed of 1m/s and also when the UAV was operating at an altitude of 2m. This report entails the final design of the Image processing subsystem of the UAVPayloadTAQ project by Group 16 and is not subjected to further modifications.

# Appendix

## Appendix A





Figure 5: Type A targets

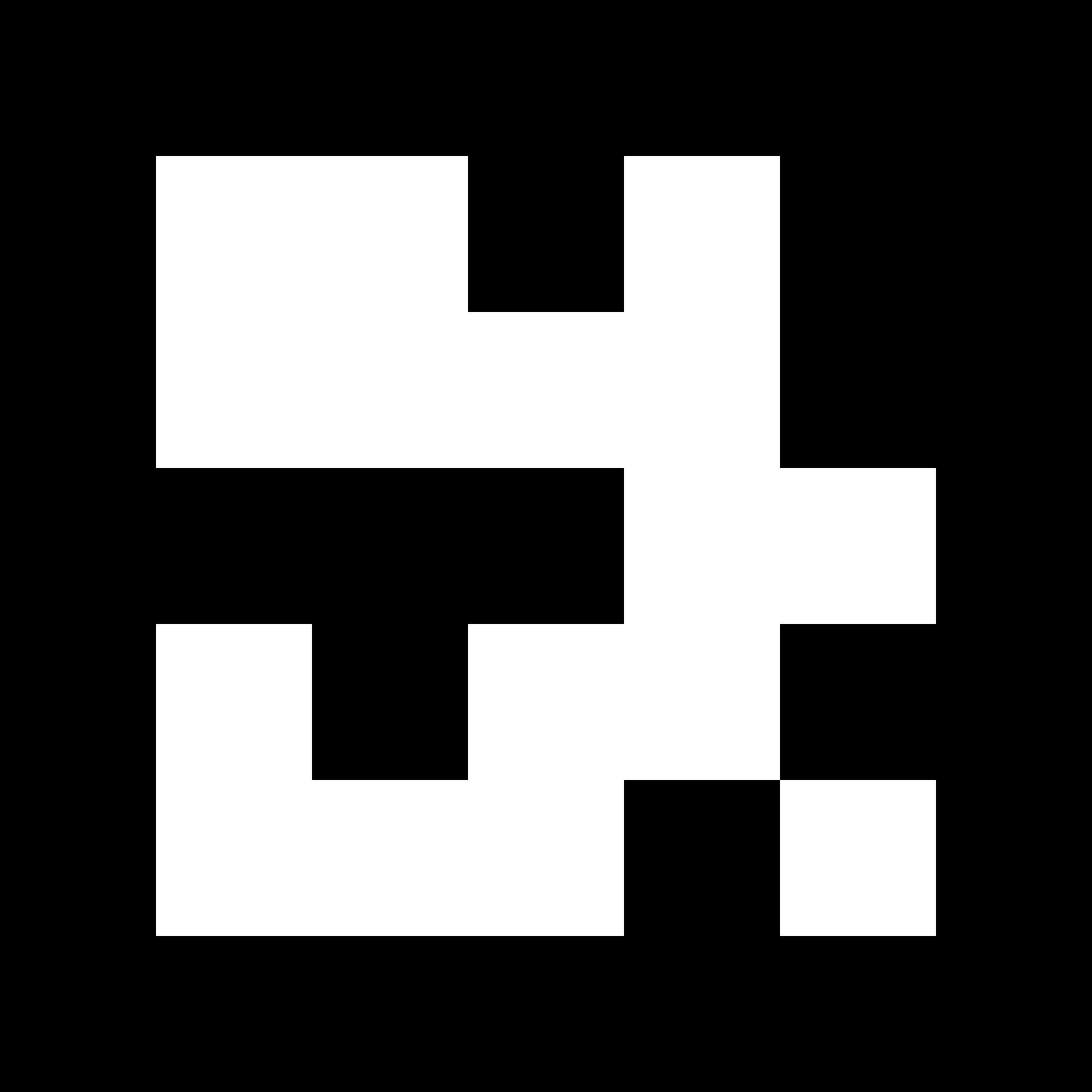


Figure 6: Type B Target

## 7.2 Appendix B

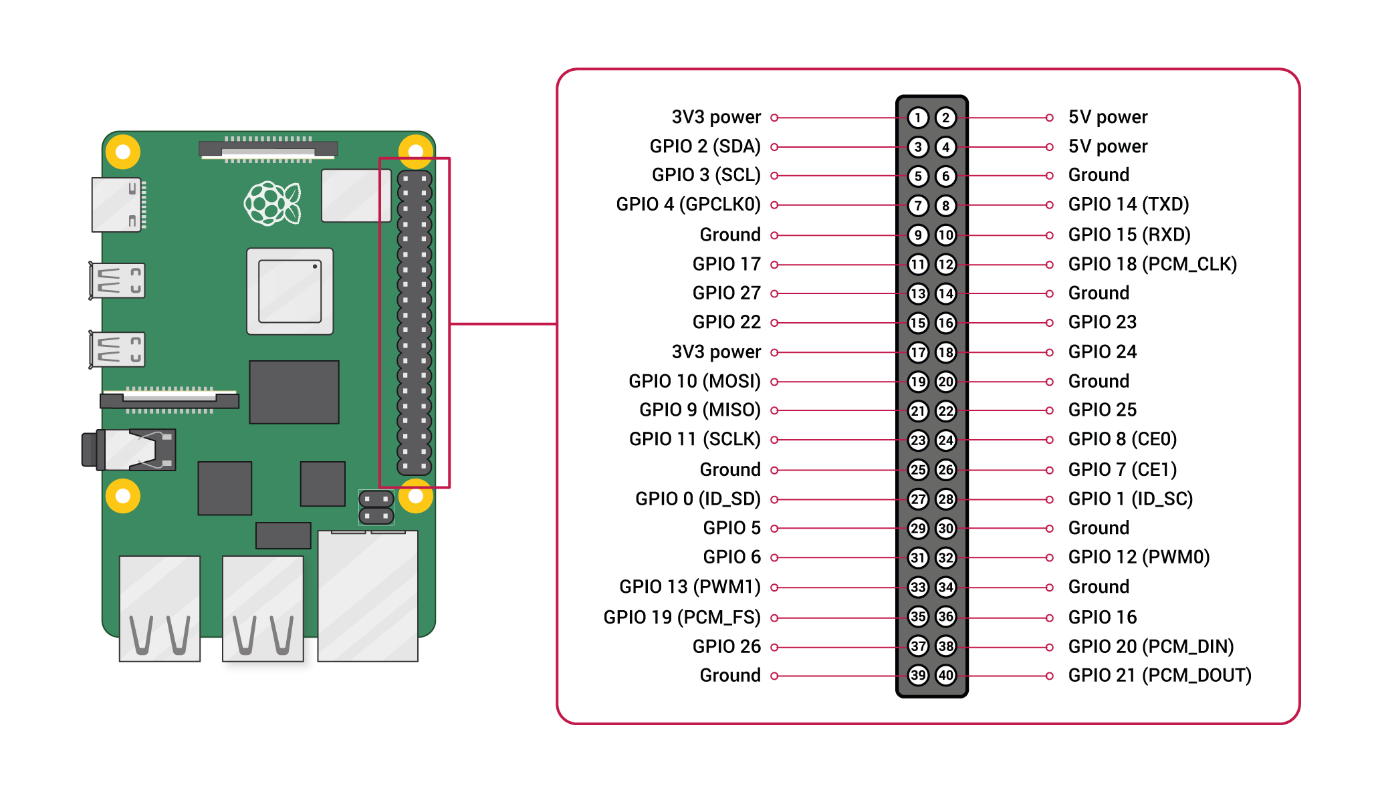


Figure 7: Raspberry Pi 3 pins layout

|  |
| --- |
|  |

## 7.3 Appendix C

**Python script for Haar Cascade Classifier:**

from \_\_future\_\_ import print\_function

import cv2 as cv

import argparse

def detectAndDisplay(frame):

frame\_gray = [cv.cvtColor](https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html#ga397ae87e1288a81d2363b61574eb8cab)(frame, cv.COLOR\_BGR2GRAY)

frame\_gray = [cv.equalizeHist](https://docs.opencv.org/3.4/d6/dc7/group__imgproc__hist.html#ga7e54091f0c937d49bf84152a16f76d6e)(frame\_gray)

#-- Detect faces

faces = face\_cascade.detectMultiScale(frame\_gray)

for (x,y,w,h) in faces:

center = (x + w//2, y + h//2)

frame = [cv.ellipse](https://docs.opencv.org/3.4/d6/d6e/group__imgproc__draw.html#ga57be400d8eff22fb946ae90c8e7441f9)(frame, center, (w//2, h//2), 0, 0, 360, (255, 0, 255), 4)

faceROI = frame\_gray[y:y+h,x:x+w]

#-- In each face, detect eyes

eyes = eyes\_cascade.detectMultiScale(faceROI)

for (x2,y2,w2,h2) in eyes:

eye\_center = (x + x2 + w2//2, y + y2 + h2//2)

radius = int(round((w2 + h2)\*0.25))

frame = [cv.circle](https://docs.opencv.org/3.4/d6/d6e/group__imgproc__draw.html#gaf10604b069374903dbd0f0488cb43670)(frame, eye\_center, radius, (255, 0, 0 ), 4)

[cv.imshow](https://docs.opencv.org/3.4/df/d24/group__highgui__opengl.html#gaae7e90aa3415c68dba22a5ff2cefc25d)('Capture - Face detection', frame)

parser = argparse.ArgumentParser(description='Code for Cascade Classifier tutorial.')

parser.add\_argument('--face\_cascade', help='Path to face cascade.', default='data/haarcascades/haarcascade\_frontalface\_alt.xml')

parser.add\_argument('--eyes\_cascade', help='Path to eyes cascade.', default='data/haarcascades/haarcascade\_eye\_tree\_eyeglasses.xml')

parser.add\_argument('--camera', help='Camera divide number.', type=int, default=0)

args = parser.parse\_args()

face\_cascade\_name = args.face\_cascade

eyes\_cascade\_name = args.eyes\_cascade

face\_cascade = [cv.CascadeClassifier](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html)()

eyes\_cascade = [cv.CascadeClassifier](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html)()

#-- 1. Load the cascades

if not face\_cascade.load([cv.samples.findFile](https://docs.opencv.org/3.4/d6/dba/group__core__utils__samples.html" \l "ga3a33b00033b46c698ff6340d95569c13)(face\_cascade\_name)):

[print](https://docs.opencv.org/3.4/df/d57/namespacecv_1_1dnn.html#a701210a0203f2786cbfd04b2bd56da47)('--(!)Error loading face cascade')

exit(0)

if not eyes\_cascade.load([cv.samples.findFile](https://docs.opencv.org/3.4/d6/dba/group__core__utils__samples.html" \l "ga3a33b00033b46c698ff6340d95569c13)(eyes\_cascade\_name)):

[print](https://docs.opencv.org/3.4/df/d57/namespacecv_1_1dnn.html#a701210a0203f2786cbfd04b2bd56da47)('--(!)Error loading eyes cascade')

exit(0)

camera\_device = args.camera

#-- 2. Read the video stream

cap = [cv.VideoCapture](https://docs.opencv.org/3.4/d8/dfe/classcv_1_1VideoCapture.html)(camera\_device)

if not cap.isOpened:

[print](https://docs.opencv.org/3.4/df/d57/namespacecv_1_1dnn.html#a701210a0203f2786cbfd04b2bd56da47)('--(!)Error opening video capture')

exit(0)

while True:

ret, frame = cap.read()

if frame is None:

[print](https://docs.opencv.org/3.4/df/d57/namespacecv_1_1dnn.html#a701210a0203f2786cbfd04b2bd56da47)('--(!) No captured frame -- Break!')

break

detectAndDisplay(frame)

if [cv.waitKey](https://docs.opencv.org/3.4/d7/dfc/group__highgui.html#ga5628525ad33f52eab17feebcfba38bd7)(10) == 27:

break

## 7.4 Appendix D

**Python script for ARUCO tag detection:**

1. import numpy as np
2. import cv2
3. import cv2.aruco as aruco
4. import sys, time, math
5. marker\_size  = 10 #- [cm]
6. Rt = np.transpose(R)
7. shouldBeIdentity = np.dot(Rt, R)
8. I = np.identity(3, dtype=R.dtype)
9. n = np.linalg.norm(I - shouldBeIdentity)
10. return n < 1e-6
11. def rotationMatrixToEulerAngles(R):
12. assert (isRotationMatrix(R))
13. sy = math.sqrt(R[0, 0] \* R[0, 0] + R[1, 0] \* R[1, 0])
14. singular = sy < 1e-6
15. if not singular:
16. x = math.atan2(R[2, 1], R[2, 2])
17. y = math.atan2(-R[2, 0], sy)
18. z = math.atan2(R[1, 0], R[0, 0])
19. else:
20. x = math.atan2(-R[1, 2], R[1, 1])
21. y = math.atan2(-R[2, 0], sy)
22. z = 0
23. return np.array([x, y, z])
24. #--- Get the camera calibration path
25. calib\_path  = ""
26. camera\_matrix   = np.loadtxt(calib\_path+'cameraMatrix\_raspi.txt', delimiter=',')
27. camera\_distortion   = np.loadtxt(calib\_path+'cameraDistortion\_raspi.txt', delimiter=',')
28. #--- 180 deg rotation matrix around the x axis
29. R\_flip  = np.zeros((3,3), dtype=np.float32)
30. R\_flip[0,0] = 1.0
31. R\_flip[1,1] =-1.0
32. R\_flip[2,2] =-1.0
33. #--- Define the aruco dictionary
34. aruco\_dict  = aruco.getPredefinedDictionary(aruco.DICT\_5X5\_100)
35. parameters  = aruco.DetectorParameters\_create()
36. #--- Capture the videocamera (this may also be a video or a picture)
37. cap = cv2.VideoCapture(0)
38. #-- Set the camera size as the one it was calibrated witxh
39. cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 640)
40. cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 480)
41. #-- Font for the text in the image
42. font = cv2.FONT\_HERSHEY\_PLAIN
43. while True:
44. #-- Read the camera frame
45. ret, frame = cap.read()
46. #-- Convert in gray scale
47. gray    = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) #-- remember, OpenCV stores color images in Blue, Green, Red
48. #-- Find all the aruco markers in the image
49. corners, ids, rejected = cv2.aruco.detectMarkers(image=gray, dictionary=aruco\_dict, parameters=parameters,
50. cameraMatrix=camera\_matrix, distCoeff=camera\_distortion)
52. # if ids is not None and ids[0] == id\_to\_find:
53. if np.all(ids is not None):
55. ret = aruco.estimatePoseSingleMarkers(corners, marker\_size, camera\_matrix, camera\_distortion)

        #-- Unpack the output, get only the first

1. rvec, tvec = ret[0][0,0,:], ret[1][0,0,:]
2. #-- Draw the detected marker and put a reference frame over it
3. print(aruco.drawDetectedMarkers(frame, corners))
4. #--- Display the frame
5. cv2.imshow('frame', frame)
6. #--- use 'q' to quit
7. key = cv2.waitKey(1) & 0xFF
8. if key == ord('q'):
9. cap.release()
10. cv2.destroyAllWindows()

        break