

《物联网与嵌入式技术》

大作业课题报告

手势识别控制柔性臂

学 生 姓 名：_____陈龙_____

学科、 专业：_____机械工程_____

学 号：_____22304034_____

完 成 日 期：_____2024. 5. 2_____

大连理工大学

Dalian University of Technology

目 录

1	课题介绍	1
1.1	背景及必要性	1
1.2	具体功能设计	1
2	MQTT 通讯	2
2.1	发布端实现	2
2.2	订阅端实现	3
3	不足与展望	4

1 课题介绍

本课题题目为“手势识别控制柔性臂”，主要内容是以摄像头作为客户端，识别手势信息作为输入，通过 MQTT 协议，发送相应手势信息控制柔性臂，实现远程控制的效果。

1.1 背景及必要性

“手势识别控制柔性臂”是一种先进的技术，结合了手势识别和柔性臂控制技术，旨在实现通过手势来控制柔性臂的运动和操作。这种技术通常应用于机器人领域，特别是在需要与人类进行直接互动的场景中，例如辅助生活机器人、医疗机器人等。

具体而言，手势识别技术用于捕捉人类手部动作和姿势，通过摄像头、传感器或其他设备来实现。这些手势被解释为命令或控制信号，然后传送给柔性臂的控制系统。柔性臂控制技术负责解释这些信号，并相应地控制柔性臂的运动，以实现预期的任务或动作。

这种技术的应用领域广泛，包括但不限于以下几个方面：

工业自动化：在工厂生产线上，通过手势来控制柔性臂可以提高生产效率和灵活性，使机器人能够根据操作人员的指示进行不同的操作，例如装配、搬运等。

辅助生活：在辅助生活机器人中，手势识别控制柔性臂可以帮助残障人士完成日常生活中的各种任务，例如拿取物品、打开门窗等。

医疗应用：手势识别控制柔性臂还可以用于医疗领域，例如在手术中协助医生进行精细的操作，或者帮助康复患者进行运动康复训练。

教育和娱乐：这项技术也可以应用于教育和娱乐领域，例如用于教学演示、虚拟现实游戏等。

1.2 具体功能设计

“手势识别控制柔性臂”共分为四个模块进行，分别为实时手势识别控制、3D 建模与动作同步、远程通讯和实物组装调试，技术路线如图 1.1 所示。

首先，实时手势识别控制部分。小组结合 OpenCV 和 Python，构建了一个高效、准确、稳定的手势控制界面，通过该界面，用户可以通过简单的手势动作实现对包括但不限于机械臂的控制，提高交互的便捷性和自然性。

3D 建模与动作同步部分。小组成员通过 Matlab 和 Unity3D 完成了对柔性臂的建模仿真。

远程通讯部分。小组以 ubuntu 系统搭建了 MQTT broker 服务器，负责接收和转发

消息。将摄像头作为发布端，负责发送手势信息，Matlab 作为订阅端，订阅手势“Gesture”话题，用以接收发布端的消息并做出相应动作。

实物组装调试部分。小组通过建模的柔性臂模型，3D 打印出实物。技术路线如下图所示。

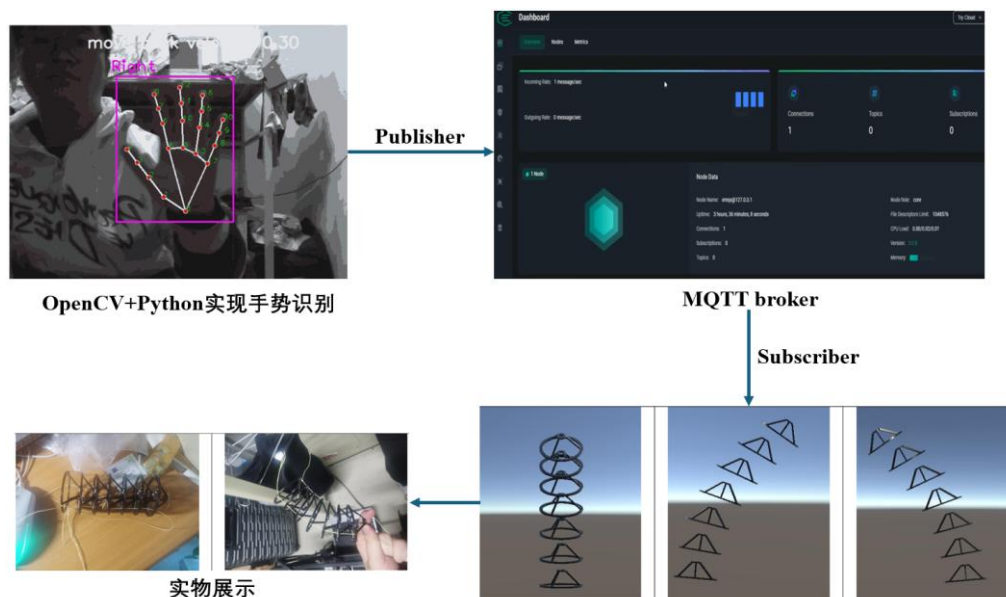


图 1 技术路线图

2 MQTT 通讯

MQTT (Message Queuing Telemetry Transport) 是一种轻量级的、灵活的、开放的、可扩展的协议，专为低带宽、高延迟或不稳定网络环境下的物联网 (IoT) 通信而设计。

MQTT 采用发布/订阅 (Publish/Subscribe) 模式，消息的发送者称为发布者 (Publisher)，消息的接收者称为订阅者 (Subscriber)。这种模式使得通信更加灵活，因为发布者和订阅者之间没有直接的联系，而是通过主题 (Topic) 进行通信，课题中使用的 IP 地址为 192.168.0.205，端口号为 1883。

2.1 发布端实现

摄像头获取手势信息，作为发布者，发布手势相对应的信息。实现部分代码如图 2。其中发布消息格式为字典，数字对应手势的动作，如图 3 所示。

```

import paho.mqtt.client as mqtt
import threading
import random
import time

# 连接成功回调
class Mqtt_Publisher:
    """
    mqtt消息通讯接口
    """
    # 此处的IP和port应该填服务器的IP和port
    def __init__(self, central_ip='192.168.0.205', port=1883, node_name='gesture', anonymous=True, timeout=60):
        """
        :param central_ip: Broker的地址
        :param port: 端口号
        :param timeout: 连接延时
        :param node_name: 节点名称
        :param anonymous: 是否同时允许多个节点
        """
        self.broker_ip=central_ip
        self.broker_port=port
        self.timeout=timeout
        self.connected=False
        self.node_name=node_name

```

图 2 发布端实现

```

# 如果连接成功, 在'gesture'的topic, 发布手势的上、下、左、右、向前、向后、停止
#intToDir = {0: "向上", 1: "向右", 2: "向下", 3: "向左", 10: "向前", 11: "向后", 12: "停止"}
pub.Publish('gesture', intToDir[curDir])
time.sleep(1)

```

图 3 消息格式

2.2 订阅端实现

Matlab 作为订阅端, 通过解析发布端的消息格式, 完成柔性臂跟随手势的相关动作, 两者以“gesture”为话题, 进行通讯。订阅端实现部分代码如图 4。

```

import paho.mqtt.client as mqtt
import threading
import random
import time
# 连接成功回调
class Mqtt_Subscriber:
    """
    mqtt消息通讯接口
    """
    # 此处的IP和port应该填服务器的IP和port
    def __init__(self, central_ip='192.168.0.205', port=1883,
                 topic_name='gesture', callback_func=None,
                 node_name='gesture', anonymous=True, timeout=60):
        """
        :param central_ip: Broker的地址
        :param port: 端口号
        :param topic_name: 接收的消息名称
        :param callback_func: 指定回调函数
        :param timeout: 连接延时
        :param node_name: 节点名称
        :param anonymous: 是否同时允许多个节点
        """

```

图 4 订阅端实现

3 不足与展望

手势识别部分识别的动作过少，增加更多手势动作，可以扩宽应用场景，以便提高适用范围。

3D 建模太过简单，并不能完全展示手势动作的完整性。

远程通讯可以借助图形工具做 UI 界面，使功能展示更加清晰。

实物组装可以根据 3D 模型做出相应完善。