**به نام خدا**

**پروژه: سطل زباله هوشمند**

**استاد:دکتر امیرمهدی منزه**

**اعضا گروه:**

**پوریا رحیمی**

**فاطمه عسکری**

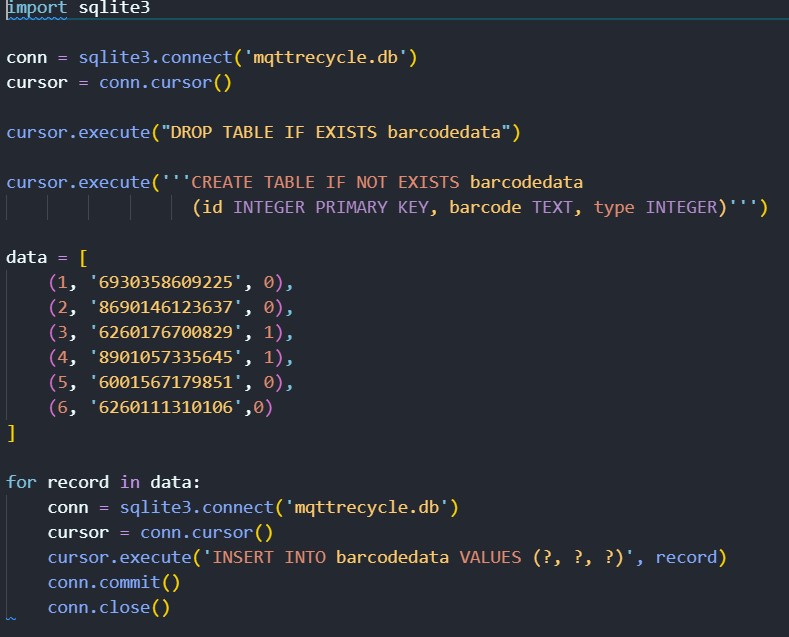
**شرح پروژه:**

هدف از انجام این پروژه این هست که به فرایند تفکیک زباله کمک کنیم در واقع این سطل زباله با توجه با اینکه یک سری زباله خاص را قبول میکند به ما کمک می کند که وقت و هزینه کمتری برای تفکیک زباله و بازیافت مصرف کنیم.

یک ماژول بارکد اسکنر داریم که بارکد کالاها را اسکن می کند سپس از طریق ESP32 به WIFI متصل میشویم و از طریق بروکر mqtt در دیتابیس بارکد کالا را سرچ میکند اگر بارکد موجود بود در سطل زباله به مدت 20 ثانیه باز میشود و می تواند زباله را درون سطل زباله بیاندازد و اگر بارکد در دیتابیس موجود نبود در سطل زباله باز نمی شود و یک LED روشن میشود.

**نحوه پیاده سازی:**

در ابتدا با استفاده از کد پایتون و sqllite دیتابیس خود را ایجاد می کنیم و چند نمونه نیز برای تست در آن insert میکنیم:



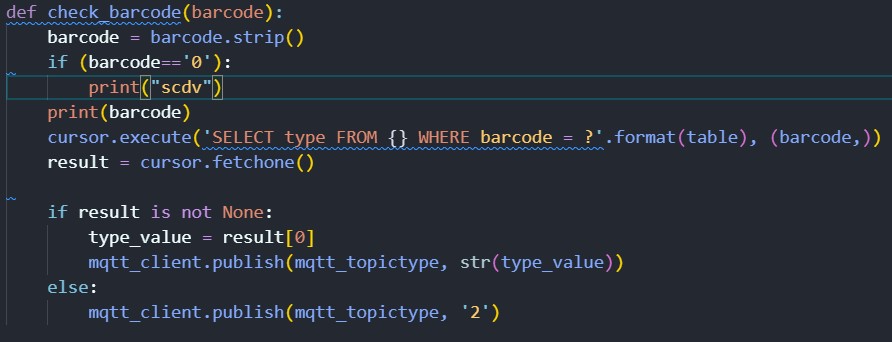
در مرحله بعد یک برنامه را با استفاده از پایتون پیاده‌سازی می‌کنیم که با استفاده از پروتکل MQTT، به یک بروکر MQTT متصل می‌شود و برای دریافت بارکد‌ها از یک دستگاه ESP32 استفاده می‌کند. سپس بارکد دریافت شده را در پایگاه داده SQLite سرچ می‌کند و نوع متناظر با هر بارکد را در یک تاپیک دیگر در بروکر MQTT منتشر می‌کند.

در ابتدا، متغیرهای مربوط به بروکر MQTT و اطلاعات اتصال به آن (آدرس بروکر، پورت، نام کاربری و رمز عبور) تعریف می کنیم:



سپس، یک اتصال به پایگاه داده SQLite برقرار می‌شود و از طریق اجرای دستورات SQL روی جدول مربوطه، بررسی و استخراج نوع متناظر با بارکد انجام می‌شود.

تابع check\_barcode ورودی بارکد را دریافت می‌کند و ابتدا بارکد را تمیز می‌کند (حذف فاصله‌های اضافی) و سپس با استفاده از یک دستور SQL، نوع متناظر با بارکد را در پایگاه داده جستجو می‌کند. اگر بارکد در پایگاه داده وجود داشته باشد، نوع متناظر با آن بارکد را از پایگاه داده خوانده و در یک موضوع MQTT منتشر می‌کند. در غیر این صورت، مقدار 2 را در موضوع MQTT منتشر می‌کند.



توابع on\_connect و on\_message نیز به ترتیب برای رسیدن به اتصال با بروکر MQTT و دریافت پیام‌ها از آن استفاده می‌شوند. در تابع on\_connect، عملیات اشتراک‌گذاری برای دریافت پیام‌های مربوط به بارکدها انجام می‌شود و تابع on\_message هنگام دریافت یک پیام جدید، بارکد را از آن استخراج کرده و به تابع check\_barcode ارسال می‌کند.

در نهایت، یک نمونه از کلاس mqtt.Client ایجاد می‌شود و توابع on\_connect و on\_message به آن اختصاص داده می‌شوند. سپس با استفاده از متدهای connect و loop\_forever، اتصال به بروکر MQTT برقرار می‌شود و برنامه در یک لوپ بی نایت منتظر دریافت پیام‌ها می‌ماند.

در واقع هر وقت که بارکد یک کالا توسط بارکد اسکنر خوانده شد esp32 در یک تاپیک بارکد کالا را publish می کند و در این کد پایتون آن را subscribe می کند و با توجه به اینکه این بارکد در دیتابیس هست یا نه پیامی را در یک تاپیک دیگر publish میکند و esp32 ان را subscribe میکند که عملیات مورد نظر را انجام دهد.

کدی که برای eesp32 در Arduino زدیم نیز به صورت زیر است:

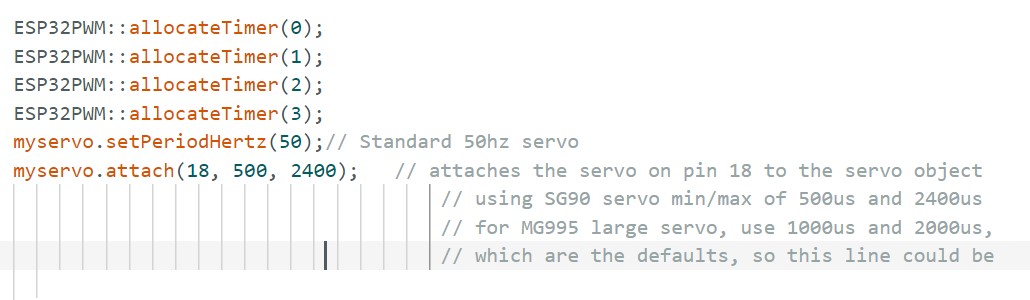
این کد با استفاده از پروتکل MQTT با یک بروکر عمومی به نام EMQX ارتباط برقرار می‌کند. برنامه شامل اتصال به شبکه Wi-Fi، ارسال و دریافت پیام‌ها از بروکر MQTT، کنترل یک سرووموتور برای تغییر جهت و چرخش، و کنترل یک LED است.

اجزای اصلی کد عبارتند از:

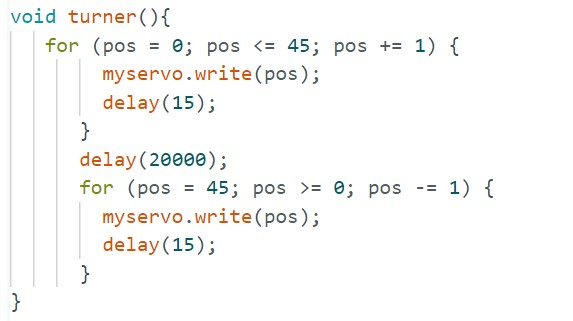
* تنظیمات شبکه :Wi-Fi شامل نام شبکه (SSID) و رمز عبور (password) است.
* تنظیمات بروکر :MQTT شامل آدرس (mqtt\_broker)، پورت (mqtt\_port)، نام کاربری (mqtt\_username) و رمز عبور (mqtt\_password) بروکر MQTT است.



* متغیرهای سرووموتور: شامل myservo (برای کنترل سرووموتور) و pos (برای زاویه فعلی سرووموتور) است.



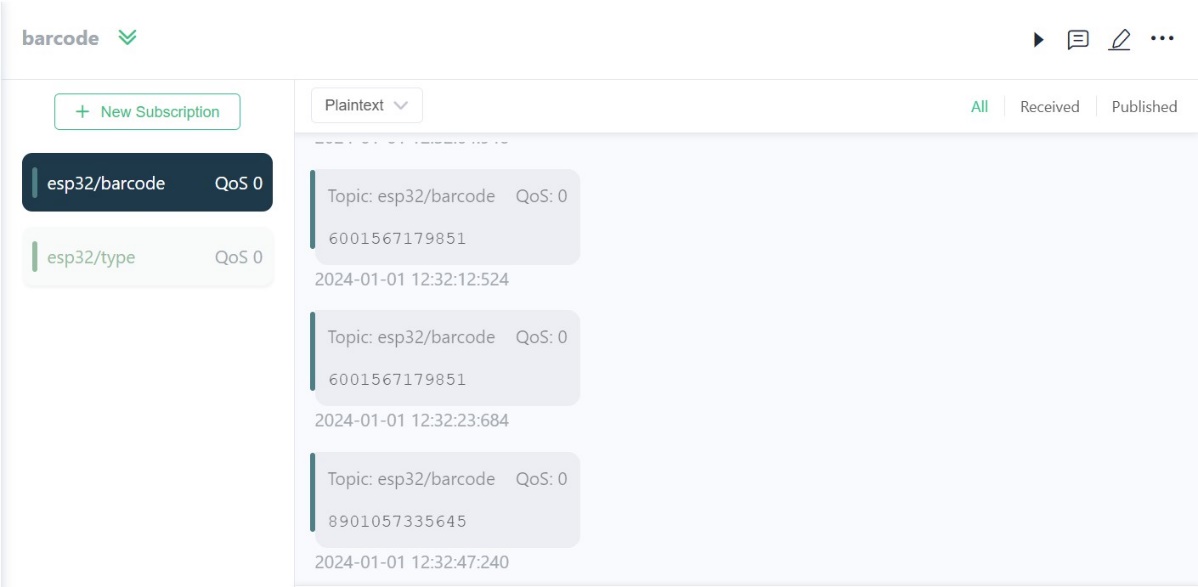
* تابع :setup شامل تنظیمات اولیه برنامه از جمله اتصال به شبکه Wi-Fi و بروکر MQTT، تنظیمات سرووموتور و تنظیمات دیگر است.
* تابع: callback برای پردازش پیام‌های دریافتی از بروکر MQTT استفاده می‌شود. این تابع به طور خاص برای کنترل سرووموتور و LED فراخوانی می‌شود.
* تابع :turner برای چرخش سرووموتور به زوایای مختلف استفاده می‌شود.



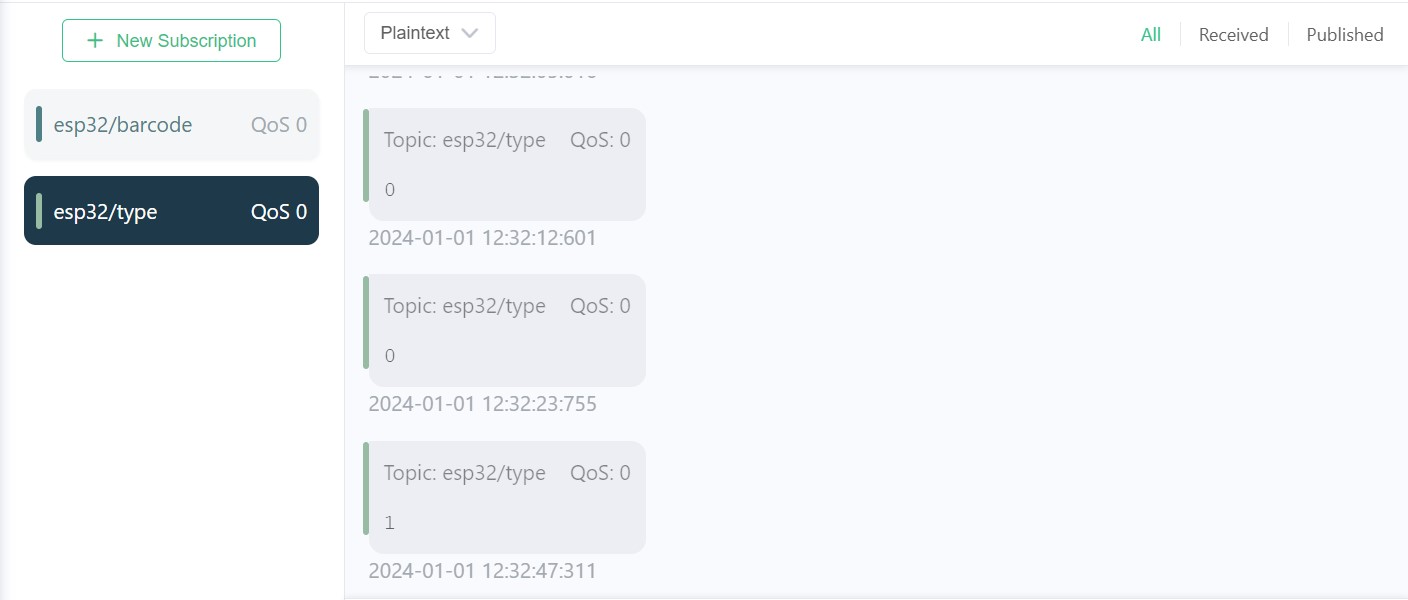
* تابع: loop برای اجرای حلقه اصلی برنامه استفاده می‌شود. در این حلقه، پیام‌های ارسالی از سنسور بارکد خوان خوانده شده و به بروکر MQTT ارسال می‌شوند. همچنین، پیام‌های دریافتی از بروکر MQTT پردازش می‌شوند.

هم چنین برای دیدن پیام هایی که publish و subscribe میشوند از برنامه MQTTX استفاده کردیم:

وقتی بارکد را می خوانیم و ان را در تاپیک esp32/barcode پابلیش می کنیم پیام ها به صورت زیر است:



و هم چنین موقعی که type کالا را در تاپیک esp32/type پابلیش می کنیم پیام ها به صورت زیر است:



سخت افزار پروژه ما نیز به صورت زیر است:

