

طراحی و ساخت یک سیستم تشخیص نوع حرکت با استفاده از ماژول esp32 و سنسورهای ژيروسکوپ سه بعدی و شبکه عصبی عمیق

هستی رادفر

کارشناسی ارشد مهندسی پزشکی دانشگاه علم و صنعت

چکیده

این مقاله درباره تشخیص نوع حرکت یک فرد است که فرد در حالت نشسته، ایستاده یا خوابیده است که این امکان از طریق 3 ماژول mpu6050 و اردوینو و esp32 به وسیله ارسال اطلاعات با wifi اتفاق می افتد که طی آن با نوشتن 2 برنامه و یک شبیه سازی و همچنین تعریف این موارد در متلب به نتایج مورد نظر رسیدیم که این نتایج شامل نمودارهایی هست که به ما اطلاعاتی درباره حرکت فرد میدهد و حتی شاید امکان پیش بینی حرکات بعدی را بتوان انجام داد. از کاربرد این دستگاه میتوان در علم حرکت شناسی انجام کارهای تحقیقاتی و آزمایشی و همچنین فیزیوتراپی نام برد و همچنین استفاده برای سالمندان و کودکان برای دانستن حرکت آنها از راه دور استفاده کرد این دستگاه را نه تنها به یک انسان حتی به حیوان و یا یک دستگاه یا ربات هم می شود متصل کرد و از آن ثبت داده کرد که دادهای آن شامل 9 مختصات X,Y,Z است که تمامی آن دسته بندی می شود و بعد به متلب داده و شبیه سازی می شود.

کلید واژه ها: سیستم تشخیص نوع حرکت، سنسورهای ژيروسکوپ سه بعدی، شبکه عصبی عمیق

مقدمه

اهمیت نوع حرکت افراد و اجسام در کار های تحقیقاتی جز موارد مهم برای بررسی اثر دارو یا یک شوک الکتریکی به مغز و یا آزمایشات دیگر هست چون حرکت افراد با اعصاب محیطی کاملاً درگیر هست و خیلی تحقیقات در این موارد بوده است پس دانستن نوع حرکت اشخاص یا حیوانات و یا روابط ها میتواند حائز اهمیت باشد.

مبانی نظری

ESP32 چیست؟

دلایل استفاده از ESP32

ESP32 محصولی از کمپانی Espressif چندین سال است که پا به عرصه بازار گذاشته است. در این سال ها به نظر می رسد ESP32 به یک ماژول سازنده ی کم هزینه برای علاقه مندان و توسعه دهندگان پروژه های اینترنت اشیا تبدیل شده است.

در طول سال گذشته شاهد استفاده از ESP32 در طراحی چندین سیستم تولیدی . همچنین رشد قابلیت ها و جامعه مخاطبان آنچنان بوده که استفاده از آن اکنون برای بسیاری از توسعه دهندگان IOT بسیار جذاب شده است. ۵ دلیل جذابیت استفاده ESP32 در اینترنت اشیا را با هم می خوانیم.

دلیل اول - ESP32 ارزان است

از این نظر، قیمت ارزان ماژول های ESP32 در مقایسه با بسیاری از راه حل های اتصال دیگر که امروزه در فروشگاه ها قابل دسترس هستند، بسیار جذاب است. قابلیت ها و جامعه کاربران ESP32 چنان رشد کرده است که ماژول را برای توسعه دهندگان پروژه اینترنت اشیا بسیار جالب می کند.

دلیل دوم - ESP32 دارای بلوتوث و Wi-Fi است

وقتی صحبت از اینترنت اشیا می شود، بحث پشتیبانی از چندین روش برای اتصال همیشه مطرح است. در برخی موارد ممکن است برای اتصال های محلی استفاده از بلوتوث را ترجیح دهیم. اما برای تعاملات ابری و از راه دور با دستگاه ها، به اتصال wifi نیاز داریم.

هنگامی که یک ماژول، بتواند از هر دو پشتیبانی کند؛ نه تنها برای توسعه دهنده بلکه برای کاربر نهایی هم یک برد محسوب می شود. اگر محصولی داشته باشیم که فقط به اتصال Wi-Fi نیاز داشته باشد. می توان گزینه ی بلوتوث را غیرفعال کرد. یا به عنوان یک ویژگی بعداً آن را اضافه نمود.

برخی از مشتریان که ESP32 را برای ویژگی های Wi-Fi خود انتخاب می کنند، برای این که مجموعه ویژگی های محصولات خود را گسترش دهند، بعداً بلوتوث را هم فعال می کنند.

دلیل سوم - ESP32 و پشتیبانی از اکوسیستم گسترده

ESP32 دارای یک اکوسیستم (جامعه کاربری) پرجمعیت است. در شروع کار با آن معمولاً از ابزارهای آزاد و یا کتابخانه های متن باز مانند کتابخانه های ارائه شده از طریق Arduino IDE استفاده می کنیم. اما محیط های توسعه حرفه ای تری نیز در دسترس هستند.

برای مثال، Espressif ابزارهایی برای توسعه و همچنین چارچوب های نرم افزاری که به اندازه ی کافی قوی به نظر می رسند را، ارائه می دهد تا بیشتر توسعه دهندگان را راضی نگه دارد.

حتی در اکوسیستم FreeRTOS آمازون از ESP32 پشتیبانی می شود. توسعه دهندگان می توانند آمازون FreeRTOS را دانلود و یک ESP32 با FreeRTOS متصل به خدمات وب آمازون (AWS) را فقط در عرض چند ساعت راه اندازی کنند.

قبول کنید این یک دلیل بسیار مهم برای استفاده از ESP32 در اینترنت اشیا می باشد.

دلیل چهارم - گزینه های Flash و PSRAM بیشتری در دسترس هستند.

دلیل چهارم برای استفاده از ESP32 در اینترنت اشیا این است که ماژول های ESP32، دارای انواع مختلف با قابلیت های متنوعی هستند.

به تجربه نشان می دهد که توانایی مقیاس پذیری بر اساس میزان فلش خارجی که در ماژول ESP32 موجود است، می تواند در میزان ذخیره سازی داده برای یک برنامه مفید باشد.

حتی در برخی از ماژول ها مانند ESP32-WROVER می توان RAM را افزایش داد. pSRAM می تواند RAM را تا ۴ مگابایت افزایش دهد و حافظه ی کافی برای هر نوع کاربردی را به ماژول بدهد

دلیل پنجم - امکان استفاده از MicroPython در ESP32

شاید تا به حال در خصوص اینترنت اشیا با میکروپایتون شنیده باشید. ESP32 یک راه حل جالب است به این دلیل که امکان نصب MicroPython در این ماژول و بنابراین توسعه به زبان Python وجود دارد.

اگرچه این روش را برای سیستم های تولیدی ترجیح نمی دهیم، اما می توانید از پایتون برای طراحی سریع نمونه اولیه (پروتوتایپ یا پیش نمون) استفاده و مشخص کنید که آیا یک پروژه اینترنت اشیا در ESP32 ممکن است یا خیر!

نصب و استفاده از MicroPython در ESP32 بسیار ساده است.

مشخصات و ویژگی های ESP32

- ✓ پردازنده ۲ هسته ای (به استثناء مدل esp32-s0wd که پردازنده تک هسته ای دارد)
 - ✓ فرکانس پردازش تا ۲۴۰ مگاهرتز
 - ✓ 448 کیلوبایت حافظه ROM همون حافظه ای که کدهای برنامه نویسی داخلش میشنه، مقایسه کنید با ۳۲ کیلوبایت حافظه ATMEGA328
 - ✓ 520 کیلوبایت حافظه SRAM
 - ✓ پشتیبانی از اتصال وایفای ۸۰۲،۱۱ با سرعت بیشینه ۱۵۰ Mb/g/n
 - ✓ پشتیبانی از بلوتوث نسخه ۲
 - ✓ پشتیبانی از بلوتوث کم انرژی BLE
 - ✓ 34 پایه قابل برنامه ریزی GPIO
 - ✓ پشتیبانی از اینترنت برای اتصال LAN
 - ✓ پشتیبانی از SDIO
 - ✓ 1 کانال اختصاصی PWM برای موتور و تا ۱۶ کانال PWM دیگر
- واقعاً این همه امکانات و ویژگی های حیرت آور در مقایسه با قیمت این میکرو نشان میدهد یکی از بهترین گزینه های برای پروژه های مختلف هست.

روش های مختلف برنامه نویسی ESP32

یک سخت افزار حرفه ای و خوب مثل ESP32 خود را محدود به یک روش برنامه نویسی یا یک کامپایلر نمیکند. محیط های رایج برنامه نویسی ESP32 شامل موارد زیر میشود:

- IO Platform ✓
- MicroPython ✓
- IDF Espressif ✓
- JavaScript ✓
- LUA ✓

برای شروع کار با ESP32 از آنجایی که پلتفرم Arduino IDE و زبان برنامه نویسی C++ یار قدیمی ماست بیشتر آموزش های حول ESP32 با این پلتفرم دنبال میشود البته مطمئن باشید از میکروپایتون هم غافل نمی شویم.

بوردهای توسعه ESP32

ماژول های مختلفی با هسته ESP32 وجود دارد که از رایج ترین آنها ESP-WROOM-32 هست که داخل ایران هم به راحتی پیدا می شود برای این ماژول برد های توسعه متفاوتی با امکانات مختلفی (مثل OLED، دوربین، شارژر باتری و ...) زده شده که این امکانات با توجه به بردی که دارید مشخص می شود

معرفی ESP32

ESP32 در واقع یک سری تراشه های میکروکنترلر است که توسط Espressif Systems در شانگهای تولید می شود. این تراشه در تعدادی از ماژول های ارزان قیمت در بازار موجود است.

ESP32 یک نسخه ی به روز شده از ESP8266 است. این برد به طرُق مختلف طراحی ESP8266 را بهبود بخشیده است. ESP32 علاوه بر قابلیت WIFI، دارای دو قابلیت بلوتوث و BLE (بلوتوث کم انرژی) نیز می باشد. همچنین قابلیت کار در حالت کم مصرف را دارد.

از دیگر ویژگی های ESP32 می توان به موارد زیر اشاره کرد:

□ ۱۸ مبدل ۱۲ بیتی آنالوگ به دیجیتال

□ ۲ مبدل دیجیتال به آنالوگ ۸ بیتی

□ ۱۰ سنسور لمسی خازنی

□ ۴ کانال SPI

□ دو رابط I2C

□ ۲ رابط I2S (برای صدای دیجیتال)

□ ۳ UART برای ارتباطات

□ ۸ کانال کنترل از راه دور IR

□ ۱۶ کانال LED PWM (مدولاسیون عرض پالس)

□ ۱ سنسور یکپارچه ی اثرهال

□ ۱ پیش تقویت کننده ی آنالوگ بسیار کم مصرف

توجه داشته باشید که بسیاری از پین های ESP32 دارای تعدادی از توابع فوق هستند. بنابراین همه ی آن ها به طور همزمان در دسترس نیستند.

ماژول های ESP32

ماژول های ESP32 زیادی برای آزمایشگران در دسترس است. در اینجا می توانید از هر یک از آن ها برای پروژه استفاده کنید. بسیاری از این برد ها دارای یک اتصال micro-USB یکپارچه هستند که برنامه نویسی را ساده می کند. بعضی از برد ها براساس تراشه ی ESP32-WROOM ساخته شده اند. برد های ESP32 DEV KIT و ESP32 NODEMCU کاملاً مشهور هستند.

یکی از برد های خانواده ی ESP به نام ESP32-CAM بسیار محبوب بوده و دارای یک دوربین فیلم برداری کوچک و یک سوکت micro SD است که با ESP32 ادغام شده است. این برد برای برنامه نویسی به آداپتور FTDI نیاز دارد.

لطفاً توجه داشته باشید که این برد ها از ویژگی های مشابه بسیار زیادی برخوردار هستند اما از نظر پین های خروجی یکسان نیستند. در آزمایشات، به جای یک عدد پین واقعی، به عملکرد پین یعنی GPIO 4 اشاره خواهیم کرد. این کار به شما این امکان را می دهد که از نوع متفاوتی از بردی که من استفاده می کنم، استفاده کنید.



تعدادی از ماژول های esp32

پین‌های ESP32

تراشه‌ی ESP32 با ۴۸ پایه با عملکردهای مختلف ارائه می‌شود. همه‌ی پین‌ها در تمام بردهای توسعه‌ی ESP32 قابل استفاده نیستند و پین‌هایی نیز وجود دارد که نمی‌توانند استفاده شوند. سوال‌های زیادی درمورد نحوه‌ی استفاده از GPIOهای ESP32 وجود دارد.

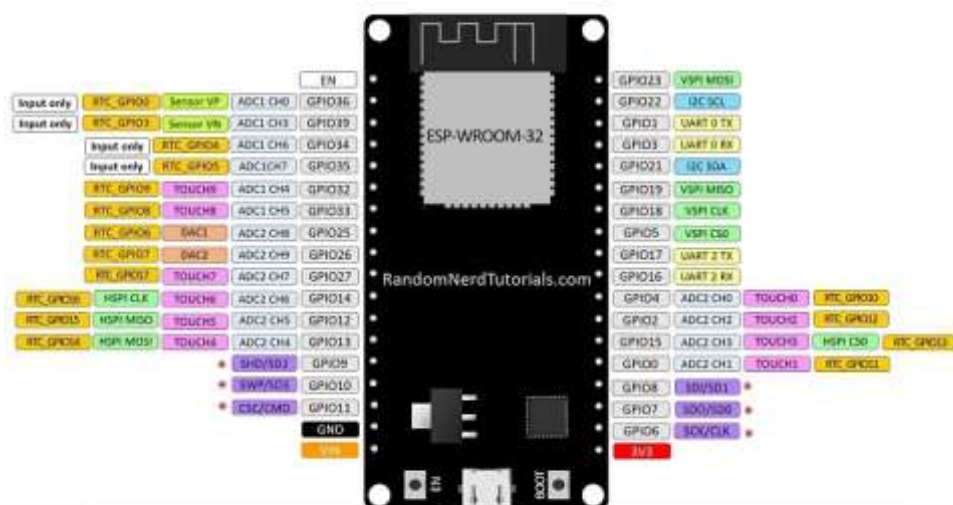
از چه پین‌هایی باید استفاده کنید؟ از چه پین‌هایی در پروژه‌های خود نباید استفاده کنیم؟

همان‌طور که در قسمت ویژگی‌ها اشاره شد، ESP32 دارای ویژگی (ADC مبدل آنالوگ به دیجیتال) و (DAC مبدل دیجیتال به آنالوگ) است. این دو ویژگی به پین‌های استاتیک خاص اختصاص می‌یابند. با این حال می‌توانیم تصمیم بگیریم که کدام پین‌ها UART، I2C، SPI، PWM و... هستند. فقط باید آن‌ها را در کد اختصاص دهید. این کار به دلیل ویژگی مالتی پلکسی‌نگ تراشه‌ی ESP32 امکان‌پذیر است. شکل زیر پین‌های مختلف یک نوع ESP32 را نشان می‌دهد.

جدول زیر نشان می‌دهد که کدام پین‌ها بهتر است به عنوان ورودی/خروجی استفاده شوند و برای احتیاط از کدام یک استفاده کنید. استفاده از پین‌های برجسته‌شده با رنگ سبز مناسب است. مواردی که با رنگ زرد برجسته‌شده‌اند استفاده از آن‌ها درست است اما باید توجه کنید که ممکن است رفتارهای غیرمنتظره‌ای هنگام BOOT شدن یا خروجی داشته باشند. پین‌های برجسته‌شده با رنگ قرمز توصیه نمی‌شود که به عنوان ورودی/خروجی استفاده شوند.

ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



پین های esp32

GPIO	ورودی	خروجی	تکته
0	Pulled up	OK	هنگام راه اندازی سیگنال PWM را خروجی می دهد.
1	OK	OK	خروجی اشکال زدایی هنگام راه اندازی
2	OK	OK	متصل به LED برد
3	OK	OK	HIGH در هنگام خروجی
4	OK	OK	
5	OK	OK	هنگام راه اندازی سیگنال PWM را خروجی می دهد.
6	OK	OK	به فلش یکپارچه SPI متصل شده است.
7	OK	OK	به فلش یکپارچه SPI متصل شده است.
8	OK	OK	به فلش یکپارچه SPI متصل شده است.
9	OK	OK	به فلش یکپارچه SPI متصل شده است.
10	OK	OK	به فلش یکپارچه SPI متصل شده است.
11	OK	OK	به فلش یکپارچه SPI متصل شده است.
12	OK	OK	در صورت بالا کشیدن BOOT از کار می افتد.
13	OK	OK	
14	OK	OK	هنگام راه اندازی سیگنال PWM را خروجی می دهد.
15	OK	OK	هنگام راه اندازی سیگنال PWM را خروجی می دهد.
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
20	OK	OK	
21	OK	OK	
22	OK	OK	
23	OK	OK	
24	OK	OK	
25	OK	OK	
26	OK	OK	
27	OK	OK	
28	OK	OK	
29	OK	OK	
30	OK	OK	
31	OK	OK	
32	OK	OK	
33	OK	OK	
34	OK	OK	نقطه ورودی
35	OK	OK	نقطه ورودی
36	OK	OK	نقطه ورودی
37	OK	OK	نقطه ورودی

رنگ بندی پین ها از سبز به قرمز

برای جزئیات بیشتر و تجزیه و تحلیل عمیق ESP32 GPIO و عملکردهای آن، به پاراگراف بعدی مراجعه فرمایید.

پین‌های فقط ورودی

GPIO های ۳۴ تا ۳۹ تنها پین‌های ورودی هستند. این پین‌ها مقاومت کششی یا کششی داخلی ندارند. نمی‌توان از آن‌ها به‌عنوان خروجی استفاده کرد. بنابراین از این پین‌ها تنها به‌عنوان ورودی استفاده کنید.

GPIO 34

GPIO 35

GPIO 36

GPIO 39

GPIOهای لمسی خازنی

قبلاً هم گفتیم که ESP32 دارای ۱۰ حسگر لمسی خازنی داخلی است. این‌ها می‌توانند تغییرات در هر چیزی را که دارای بار الکتریکی است، مانند پوست انسان احساس کنند. بنابراین آن‌ها می‌توانند تغییرات ناشی از لمس GPIO ها با انگشت را تشخیص دهند. این پایه‌ها می‌توانند به راحتی در لنت‌های خازنی ادغام شوند و جایگزین دکمه‌های مکانیکی شوند. حسگرهای لمسی خازنی به این GPIO ها متصل هستند:

T0 (GPIO4)

T1 (GPIO0)

T2 (GPIO2)

T3 (GPIO15)

T4 (GPIO13)

T5 (GPIO12)

T6 (GPIO14)

T7 (GPIO27)

T8 (GPIO33)

T9 (GPIO32)

مبدل آنالوگ به دیجیتال (ADC)

ESP32 دارای ۱۸ کانال ۱۲ بیتی ورودی ADC است. این GPIO ها هستند که می‌توانند به‌عنوان ADC و کانال‌های

مربوطه استفاده شوند:

ADC1_CH0 (GPIO 36)

ADC1_CH1 (GPIO 37)

ADC1_CH2 (GPIO 38)

ADC1_CH3 (GPIO 39)

ADC1_CH4 (GPIO 32)

ADC1_CH5 (GPIO 33)

ADC1_CH6 (GPIO 34)

ADC1_CH7 (GPIO 35)

ADC2_CH0 (GPIO 4)

ADC2_CH1 (GPIO 0)

ADC2_CH2 (GPIO 2)

ADC2_CH3 (GPIO 15)

ADC2_CH4 (GPIO 13)

ADC2_CH5 (GPIO 12)

ADC2_CH6 (GPIO 14)

ADC2_CH7 (GPIO 27)

ADC2_CH8 (GPIO 25)

ADC2_CH9 (GPIO 26)

توجه: هنگام استفاده از WIFI نمی‌توان از پین‌های ADC2 استفاده کرد. بنابراین، اگر از WIFI استفاده می‌کنید و در تهیه‌ی مقدار از DC2 GPIO، می‌توانید به جای آن از ADC1 GPIO استفاده کنید. این کار می‌تواند مشکل را حل کند. کانال‌های ورودی ADC دارای وضوح ۱۲ بیتی هستند. این بدان معناست که می‌توانید قرائت‌های آنالوگ از ۰ تا ۴۰۹۵ را به‌دست آورید که در آن ۰ مربوط به ۰ ولت و ۴۰۹۵ مربوط به ۳/۳ ولت است.

مبدل دیجیتال به آنالوگ (DAC)

برای تبدیل سیگنال‌های دیجیتال به خروجی سیگنال ولتاژ آنالوگ، ۲ کانال ۸ بیتی در ESP32 وجود دارد. کانال‌های DAC به‌صورت زیر هستند:

DAC1 (GPIO25)

DAC2 (GPIO26)

GPIOهای RTC

پشتیبانی از RTC GPIO در ESP32 وجود دارد. GPIOهایی که به زیر سیستم کم مصرف RTC هدایت می‌شوند، می‌توانند هنگام خواب عمیق از ESP32 استفاده شوند. هنگام کار با پردازشگر فوق‌العاده کم مصرف (ULP)، می‌توان از این GPIOهای RTC برای بیدار کردن ESP32 از حالت خواب عمیق استفاده کرد. GPIOهای زیر را می‌توان به‌عنوان منبع بیدار خارجی استفاده کرد.

RTC_GPIO0 (GPIO36)

RTC_GPIO3 (GPIO39)

RTC_GPIO4 (GPIO34)

RTC_GPIO5 (GPIO35)

RTC_GPIO6 (GPIO25)

RTC_GPIO7 (GPIO26)

RTC_GPIO8 (GPIO33)

RTC_GPIO9 (GPIO32)

RTC_GPIO10 (GPIO4)

RTC_GPIO11 (GPIO0)

RTC_GPIO12 (GPIO2)

RTC_GPIO13 (GPIO15)

RTC_GPIO14 (GPIO13)

RTC_GPIO15 (GPIO12)

RTC_GPIO16 (GPIO14)

RTC_GPIO17 (GPIO27)

PWM

کنترل LED PWM در ESP32 توسط ۱۶ کانال مستقل انجام می‌شود که می‌تواند برای تولید سیگنال‌های PWM با ویژگی‌های مختلف پیکربندی شود. همه‌ی پین‌هایی که می‌توانند به‌عنوان خروجی عمل کنند، می‌توانند به‌عنوان پین‌های PWM استفاده شوند. GPIOهای ۳۴ تا ۳۹ نمی‌توانند PWM تولید کنند. برای تنظیم سیگنال PWM، باید این پارامترها را در کد تعریف کنید:

Signal's frequency;

Duty cycle;

PWM channel;

GPIO جایی که می‌خواهید سیگنال را خارج کنید.

I2C

ESP32 دارای ۲ کانال I2C است و هر پایه را می‌توان به‌عنوان SDA یا SCL تنظیم کرد. هنگام استفاده از ESP32 با Arduino IDE، پایه‌های پیش‌فرض ۱۲ عبارتند از:

GPIO 21 (SDA)

GPIO 22 (SCL)

اگر می‌خواهید از پین‌های دیگر استفاده کنید، هنگام استفاده از کتابخانه‌ی `wire`، تنها باید فراخوانی کنید

PI

به‌طور پیش‌فرض، پین‌های **SPI** به شرح زیر می‌باشند:

SPI	MOSI	MISO	CLK	CS
VSPi	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

Interrupt

تمامی GPIO ها می‌توانند به‌عنوان وقفه پیکربندی شوند.

Enable (EN)

Enable (EN) پین فعال‌کننده‌ی تنظیم‌کننده‌ی ۳/۳ ولت است. زمانی که در حالت **Pulled up** است، به زمین متصل می‌شود تا تنظیم‌کننده‌ی ۳/۳ ولت را غیرفعال کند. به این معنی است که شما می‌توانید از این پین متصل به یک دکمه برای شروع مجدد **ESP32** خود استفاده کنید.

برنامه‌نویسی ESP32

ESP32 را می‌توان با استفاده از محیط‌های مختلف توسعه برنامه‌ریزی کرد. کد را می‌توان به زبان **C++** مانند آردوینو یا **Micro Python** نوشت.

برای مبتدیان یک راه آسان برای شروع، استفاده از **Arduino IDE** است. گرچه این لزوماً بهترین محیط برای کار با **ESP32** نیست اما این مزیت را دارد که یک برنامه‌ی کاربردی آشناست. بنابراین منحنی یادگیری مسطح می‌شود. ما نیز برای پروژه‌ی خود از **Arduino IDE** استفاده خواهیم کرد

روش تحقیق

اتصال **esp32** به اینترنت و بعد اتصال آن به **mpu6050** و در نهایت برنامه نویسی در آردوینو و گرفتن داده مد نظر و تبدیل آن به فایل **CSV** و با اکسل و در ادامه خروجی های مد نظر از متلب و تشخیص نوع حرکت سوژه

اهداف پژوهش**هدف کلی**

تشخیص نوع حرکت فرد یا جسم یا حیوان از طریقژیروسکوپ و **esp32**

اهداف جزئی

- تشخیص نوع حرکت سالمندان
- تشخیص نوع حرکت کودکان
- استفاده آن روی گوشی های هوشمند

پرسش‌های پژوهش

پرسش‌های اساسی یا فرضیه تحقیق

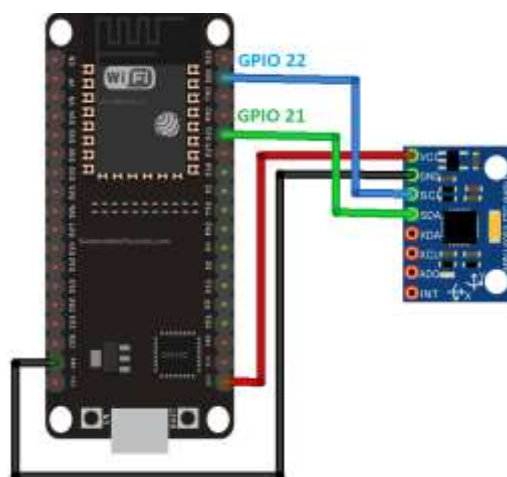
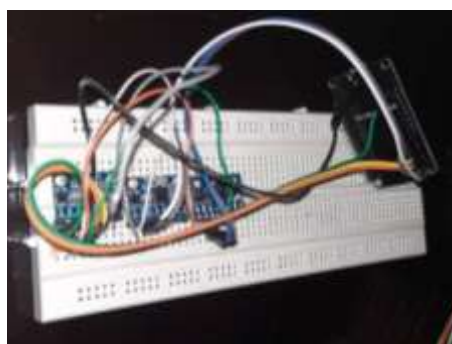
پرسشها یا فرضیه های پژوهش در اینجا آورده می شوند. هر یک از پرسشهای اصلی یا فرضیه های پژوهش، باید بر اساس هر یک از اهداف فرعی پژوهش تدوین شوند.

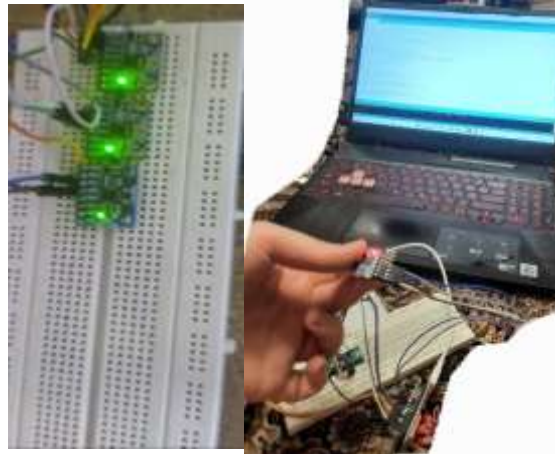
۱. اتصال esp32 به wifi به چه صورت هست
۲. ارسال داده از اردوینو به متلب چگونه هست
۳. پردازش نمودار های خروجی متلب برای رسیدن به پاسخ از چه طریقی هست

شماتیک کلی مدار به همراه خود دستگاه طراحی شده

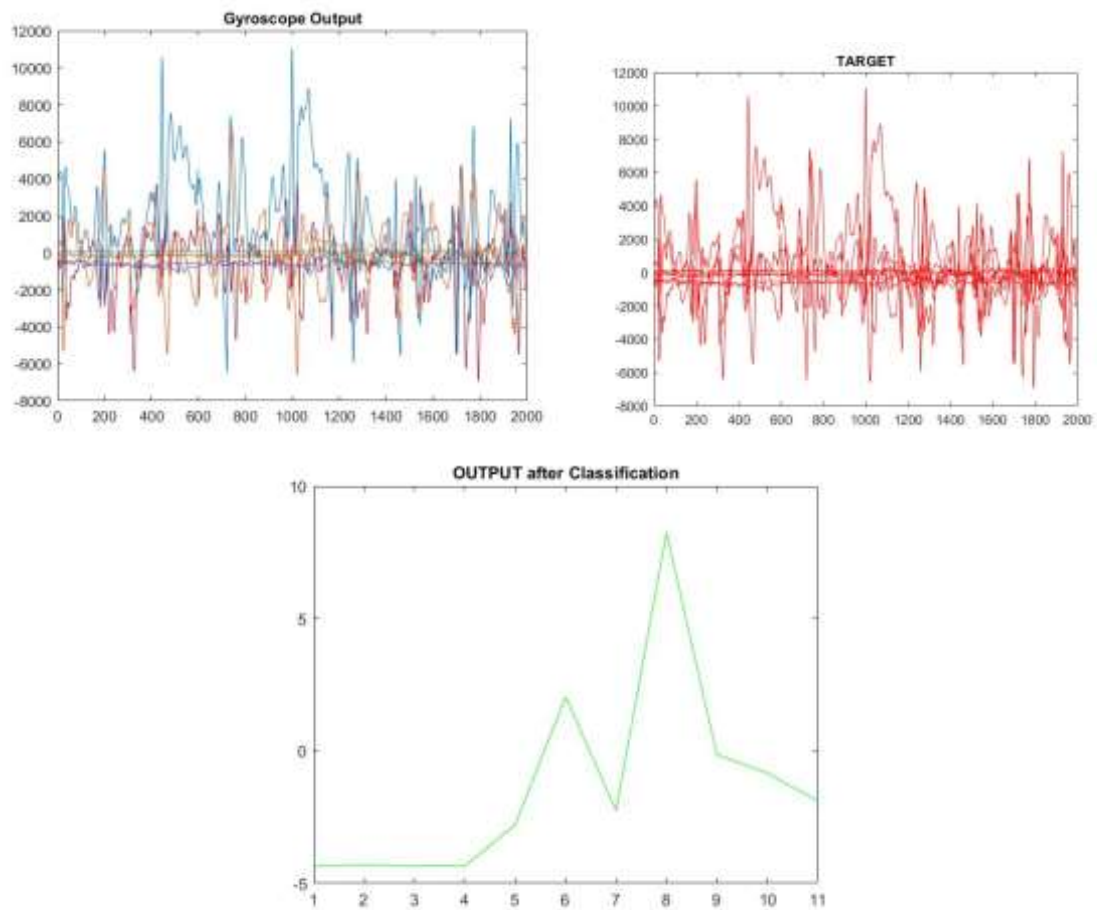
از طریق شماتیک کلی مدار میتوانیم به این اطلاعات برسیم که باید کدام پایه های میکرو و ژيروسکوپ را به هم وصل کنیم که بتوانیم داده بگیریم

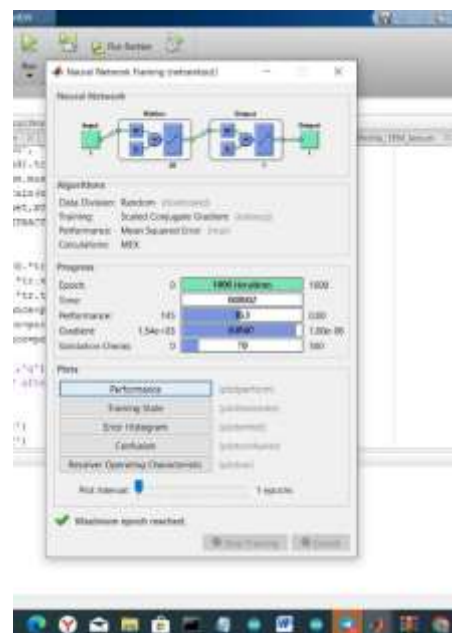
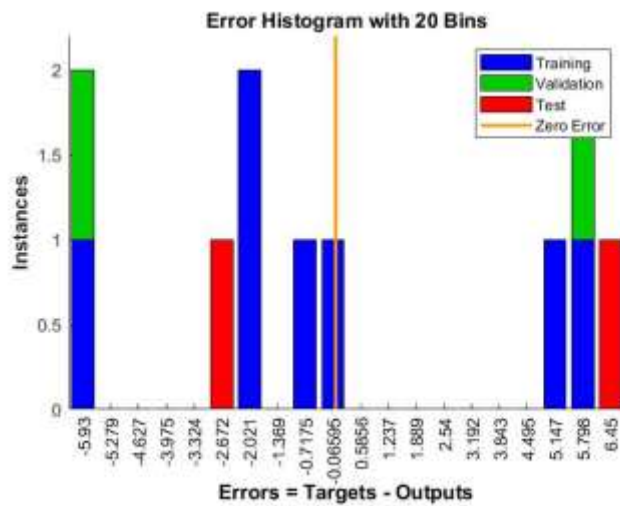
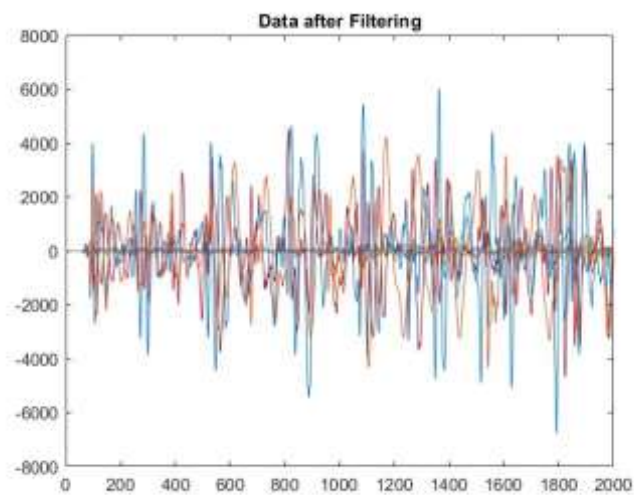
همانطور که میدانید برای تغذیه ژيروسکوپ ها و خود esp32 باید esp را به کامپیوتر وصل کرد و بعد از اتصال ان به کامپیوتر از پایه vcc ژيروسکوپ به esp وصل کنیم و همینطور زمین ها هم فراموش نشود در مرحله بعد وقتی از اتصال esp به wifi اطمینان حاصل شد پایه های sda و sci که قبلا به صورت کامل توضیح داده شد را به gpio21 و gpio22 وصل کنیم حال ما چون ۳ ژيروسکوپ داریم میتوانیم بقیه پایه ها را در esp32 درگیر کنیم ولی در mpu6050 فقط پایه های sda و sci و بعد از اتصال مرحله بعدی باید سراغ برنامه اردوینو برویم و دیتا گیری را آغاز کنیم و بعد از گرفتن دیتا و تبدیل آن به فایل csv به سراغ برنامه متلب میرویم و train کردن و همینطور learning را شروع میکنیم





خروجی متلب







ماتریس کانکلوزن

نتیجه گیری

پیشنهادهای که میتوان داشت این هست که ما میتوانستیم خروجی هارا در یک مانیتور روی برد برد نشان دهیم حتی دیتا های X,Y,Z که باعث خوانا بودن و راحتی کار در قسمت برنامه اردوینو برایمان بود.

منابع فارسی

۱. رشید , طارق , "شبکه عصبی مخصوص به خودتان را بسازید"
۲. عباسی , رضا , "درآمدی بر شبکه های عصبی"
۳. سامانه اطلاع رسانی وزارت علوم: اخبار علمی، تحقیقاتی و فناوری (۱۳۸۶). دسترسی از طریق:
۴. <http://www.msrt.ir/sites/ravabetomomi/msrtnews/lists/jsj/dispForm.spa>.

منابع لاتین

1. Google Deep Learning Community
2. <http://networkflow.net>