

صورت پروژه: پارکینگ هوشمند

تکنیک های بکار رفته در پروژه:

استفاده از پروتکل های ارتباطی **http** و **mqtt**

استفاده از **node red** بعنوان سرور

استفاده از سنسور **ultrasonic** و **servo motor**

استفاده از دیتابیس

طراحی رابط گرافیکی زیبا همراه با انیمیشن (صفحه وب)

استفاده از **html**، **css**، **java script** و **php** در طراحی صفحه وب

استفاده از هاست برای صفحه وب

استفاده از **long polling** در صفحه وب

توضیحات مربوط به نود رد:

در این پروژه **node red**، بعنوان سرور مورد استفاده قرار گرفته است. در **flow** نودرد ما، وضعیت سنسور پارک، از طریق نود **mqtt** که روی تاپیک موردنظر **subscribe** شده است، دریافت شده و از طریق یک درخواست **http**، به دیتابیس ما روی هاست ارسال می شود (شماره اسلات و وضعیت آن، به فایل **BE2.php** که در هاست ما قرار دارد ارسال می شود و از آنجا درون دیتابیس ذخیره می شود) و روی صفحه نمایش داده می شود. وضعیت سنسور گیت نیز به همین شکل است. با این تفاوت که در بکند، علاوه بر ذخیره وضعیت در دیتابیس، چک می شود که آیا در پارکینگ جای پارک خالی وجود دارد یا نه. اگر وجود داشت، 1 را برمی گرداند و این 1 با استفاده از نود **mqtt**، روی تاپیک مشخص شده منتشر می شود و در برد دریافت می شود و گیت ما در وضعیت 90 درجه قرار می گیرد. وقتی هم که سنسور ما وضعیت خالی را نشان داد، 0 برگردانده می شود و گیت در وضعیت 0 درجه قرار می گیرد. (در صورتی که در پارکینگ جای خالی وجود نداشت، فلگ ما همچنان مقدار صفر خواهد داشت و گیت در وضعیت 0 درجه باقی خواهد ماند)

توضیحات مربوط به برنامه آردوینو:

در ابتدای برنامه آردوینو کتابخانه های مورد نیاز ما در پروژه قرار گرفته اند. (ESP8266WiFi.h: کتابخانه مربوط به خود برد، Ticker.h: کتابخانه مربوط به ایجاد وقفه، PubSubClient.h: کتابخانه مربوط به استفاده از mqtt و publish و subscribe ، Servo.h: کتابخانه سروو موتور)

در این برنامه از دو تابع ultrasonic و ultrasonic2 برای چک کردن وضعیت دو سنسور اولتراسونیک ما استفاده شده است که در آن، فاصله مانع تا سنسور دریافت شده، به سانتی متر تبدیل شده و روی یک تاپیک خاص publish می شود. (این مقادیر به بروکر ارسال شده و در آنجا میزان فاصله بررسی می شود و پر یا خالی بودن اسلات ها مشخص می شود). برای فراخوانی این توابع نیز از وقفه استفاده شده است (هر 3 ثانیه) تا در بخش loop، delay استفاده نشود و بررسی وضعیت سنسور ها به طور مرتب انجام پذیرد.

برای کنترل سروو موتور نیز روی تاپیک smartparking/gate، subscribe صورت گرفته است تا اگر روی این تاپیک، بروکر مقداری را منتشر کرد، آن مقدار دریافت شده و به سروو موتور ما دستور چرخش داده شود. در تابع callback نیز بررسی شده که آیا روی این تاپیک مقداری ارسال شده یا خیر. اگر این مقدار برابر با 1 بود به سروو موتور دستور قرار گرفتن در وضعیت 90 درجه و اگر صفر بود، قرار گرفتن در وضعیت صفر را دستور می دهد. علاوه بر این ها توابع مربوط به اتصال به وای فای و سایر توابعی که در درس توضیحات مربوط به آن ها داده شده بود مورد استفاده قرار گرفته است.

توضیحات مربوط به صفحه وب:

با توجه به موضوع پروژه، طراحی یک ui که بتوان وضعیت پارکینگ را در آن مشاهده کرد یکی از مهمترین قسمت های پروژه است. این وبسایت را میتوان در آدرس <http://hamedhost.cloudsite.ir/index1.html> با ظاهری متفاوت تر از آنچه در سایر پارکینگ ها وجود دارد، مشاهده کرد. در طراحی این صفحه از html، css، java script، jquery، php و علاوه بر آن، از یک دیتابیس برای ذخیره آخرین وضعیت پارکینگ (وضعیت جای پارک ها و وضعیت گیت) استفاده شده است.

درون این صفحه علاوه بر نمایش وضعیت جای پارک ها و وضعیت گیت، بخش هایی مثل نشان دادن مکان پارکینگ روی مپ و نشان دادن وضعیت آماری پارکینگ با توجه به داده های ذخیره شده در دیتابیس (این بخش فعلا فقط به صورت گرافیکی رسم شده است) وجود دارد که میتوان آن ها را در آدرس ذکر شده مشاهده کرد. معمولا در طراحی وبسایت ها، درون صفحه رخدادی مثل فشردن یک کلید یا انتخاب یک گزینه رخ می دهد و درخواست به بکند ارسال شده و نتیجه نمایش داده می شود. اما در این جا چون وضعیت پارکینگ از طریق سنسورها بررسی می شود و با تغییر وضعیت، نتیجه باید در صفحه نمایش داده شود، کار متفاوت است. در این جا میتوان از روش های متفاوتی استفاده کرد. مثلا میتوان از سوکت پروگرامینگ استفاده کرد و پورتی را به این کار اختصاص داد تا همیشه روی این پورت گوش بدهد و با تغییر وضعیت سنسور نتیجه در صفحه نمایش داده شود. اما در این جا با توجه به پیچیدگی این روش، از یک روش دیگر به نام Long Polling استفاده شده است. به این صورت که در یک حلقه همیشگی وضعیت دیتابیس بررسی می شود و چنانچه مقدار جدیدی در آن نوشته شد، به صفحه ارسال می شود. بخشی از پیاده سازی این روش در BE.php و بخشی از آن در فایل جاوااسکریپت انجام شده است. BE2.php نیز برای رسیدگی به درخواست های ارسال شده از نودرد نوشته شده است. سایر بخش های مربوط به صفحه وب (index.html، style.css و sc.js) بیشتر مربوط به ظاهر وبسایت بوده و توضیح آن ها خارج از مجال است.