



**طرح: رابط کاربری گرافیکی ربات فیزیو**

Graphical User Interface for Physio Robot

شماره طرح نیماد: **962297**

عنوان گزارش:

Graphical User Interface for Physio Robot

05/09/1400

1. **اهداف گزارش:** با توجه به انتقال تمام کدها از متلب به پایتون و احساس نیاز به یک رابط کاربر گرافیکی، تصمیم بر آن شد که یک GUI با استفاده از PyQt طراحی شود. همچنین از آنجا که برنامه در Raspberry pie اجرا می‌شود، این فریم ورک گزینه بسیار مناسبی می‌باشد.
2. **Key Words:** GUI, Graphical User Interface, Physio Robot, Python, PyQt
3. **چکیده:**

امروزه در صنعت به همراه هر محصول که در تعامل با انسان است، نیاز به یک رابط کاربری مختص آن محصول(ربات، دستگاه‌های اندازه‌گیری و ...) داریم. هدف اصلی یک GUI، آسان‌تر کردن و سرعت دادن به فرایند کار با دستگاه است و به همین خاطر کاربرپسند (User Friendly) بودن برنامه و همچنین سرعت اجرا (Performance) آن اهمیت بالایی دارد. به همین علت در شرکت‌های مختلف به این بخش توجه زیادی می‌شود و همیشه سعی بر آن است تا برنامه با کمترین مشکل در اختیار کاربر قرار بگیرد و در آینده قابلیت‌های جدیدی به آن افزوده شود. ربات فیزیوتراپ با توجه به اهداف درمانی و توان‌بخشی که دارد و همچنین نیاز به ارتباط موثر دکتر یا مسئول دوره درمانی با ربات و ربات با بیمار، به یک رابط کاربری گرافیکی نیاز دارد تا دستورات دکتر را دریافت و تحلیل حرکات بیمار را نمایش دهد. این رابط کاربری از بخش‌های مختلف شامل صفحاتی برای حرکات Passive، Isometric، Isotonic، Isokinetic، Spring form و Water form است. برای طراحی این GUI از تمپلیت PyDracula (لینک GitHub در پیوست) استفاده شده است. این GUI با استفاده از PyQt و PySide نوشته شده است که امکانات بسیار خوبی برای طراحی یک رابط کاربری در اختیار ما قرار می‌دهند. برای مثال، با استفاده از Pyside6-designer (نسخه Pyside ممکن است متفاوت باشد) می‌توان UI برنامه را طراحی کرده و از آن خروجی Python یا C++ گرفت. در ادامه به بخش‌های مختلف کد و محیط GUI خواهیم پرداخت.

1. **موضوعات محتوایی:**

پای دراکولا (PyDracula) یک نمونه آماده رابط کاربری گرافیکی است که توسط Wanderson M.Pimenta  
نوشته و در اختیار عموم قرار داده شده است. این GUI با استفاده از Python 3.9 و PySide6 نوشته شده و استفاده از ورژن‌های قبلی ممکن است باعث ایجاد مشکلاتی در برنامه شود.

بخش‌های مختلف این GUI کاملا از هم جدا شده و به صورت ماژول Module نوشته شده‌اند که امکان تغییر و ارتقای برنامه را برای ما راحت‌تر می‌کند.

**PySide**

PySide ابزاري است براي استفاده از Qt در پايتون. مشابه PySide، PyQt را داریم که در 99 درصد موارد کاملا مشابه هستند و اصلی‌ترین تفات آنها بحث لایسنس است. Qt فريم ورکي براي ايجاد رابط گرافيکي (GUI) است که توسط شرکت نوکيا براي بکارگيري در محصولات نرم‌افزاري اين شرکت چه براي رايانه‌هاي شخصي و چه تلفن‌هاي هوشمند پشتيباني مي‌شود. Qt مولتي پلت فرم مي‌باشد پس تمامي برنامه‌هاي نوشته شده با کيوتي و پايتون بر روي کليه‌ي سيستم عامل‌ها اجرا خواهد شد.

**ماژول‌ها**  
PySide شامل زيرمجموعه‌اي از ماژول‌هاي مختلف مي‌باشد که هر بخشي وظيفه خاصي را بر عهده دارد. از بخش‌هاي مهم PySide (یا PyQt) مي‌توان موارد زير را نام برد:

QtCore: اين ماژول شامل کلاس‌های غير گرافيکي بوده و به عبارتي ديگر هسته مرکزي مي باشد که بخش‌هاي مهمي چون حلقه وقايع (event loop) و مکانيسم سيگنال و اسلات (Signal/Slot) را شامل مي شود.  
QtGui: اين بخش شامل همه کلاس‌هاي گرافيکي مي‌باشد.  
QtNetwork: اين ماژول شامل کلاس‌هايي براي ساخت برنامه‌هاي سمت سرور يا کلاينت (سرويس دهنده و سرويس گيرنده) بر پايه پروتکل هاي TCP يا UDP مي باشد. همچنين اين بخش شمال کلاس‌هاي کلاينت FTP، HTTP و همچنين جستجوي DNS مي‌باشد .  
QtOpenGL: اين ماژول امکان ايجاد و کار بر روي تصاوير سه بعدي با استفاده از اپن جي ال را فراهم مي‌کند.  
QtScript: با استفاده از اين ماژول برنامه‌هاي نوشته شده قادر به اجرا اسکريپت‌هاي جاوا خواهند بود. در حقيقت با استفاده از اين روش امکان توسعه و اسکريپت نويسي براي کاربران فراهم مي‌شود.  
QtSql: وظيفه اين ماژول ايجاد ارتباط با پايگاه داده‌هاي مبتني بر SQL مي‌باشد. همچنين اين ماژول شامل مدير ديتابيس SQLite مي‌باشد.  
QtSvg: اين ماژول شامل توابعي براي نمايش فايل‌هايي با فرمت SVG يا Scalable Vector Graphics است.  
QtTest: اين ماژول شامل توابعي براي اعمال تست بر روي اجزا برنامه نوشته شده با پايتون و کيوتي مي‌باشد.  
QtXml: اين ماژول شامل کلاس هايي مي باشد که رابط‌هاي SAX و DOM را براي تجزيه‌کننده XML کيوتي فراهم مي‌کنند.  
QtAssistant‌: با استفاده از اين ماژول مي‌توان نرم افزار Qt Assistant را در برنامه نوشته شده با پايتون و کيوتي دروني سازي کرده و در نتيجه مستندات برنامه را به همراه آن و بدون نياز به ايجاد بخشي جداگانه در خود برنامه دروني‌سازي نمود.  
QtDesigner: توسط اين ماژول امکان توسعه طراح کيوتي توسط PyQt فراهم خواهد شد. براي مثال مي‌توان افزونه يا يک عنصر جديد در PyQt ايجاد و از آن همانند ساير اشيا گرافيکي (Widget) در محيط طراح استفاده کرد.  
QAXContainer: با استفاده از اين ماژول برنامه نوشته شده قادر خواهد بود به اشيا COM و اکتيو ايکس (ActivX) ارتباط برقرار کند.  
Qt: اين ماژول شامل تمامي موارد ذکر شده در بالا مي باشد. با اضافه کردن اين ماژول ديگر نيازي نيست برنامه‌نويس ماژول شامل تابع مورد نياز خود را بداند. از معايب اين روش لود شدن همه فريم ورک کيوتي مي‌باشد که باعث اشغال زياد حافظه مي‌گردد.  
uic: اين ماژول شامل کلاس‌هايي براي کار با فايل‌هاي ui مي‌باشد که توسط طراح (Designer) کيوتي ايجاد مي‌گردد.

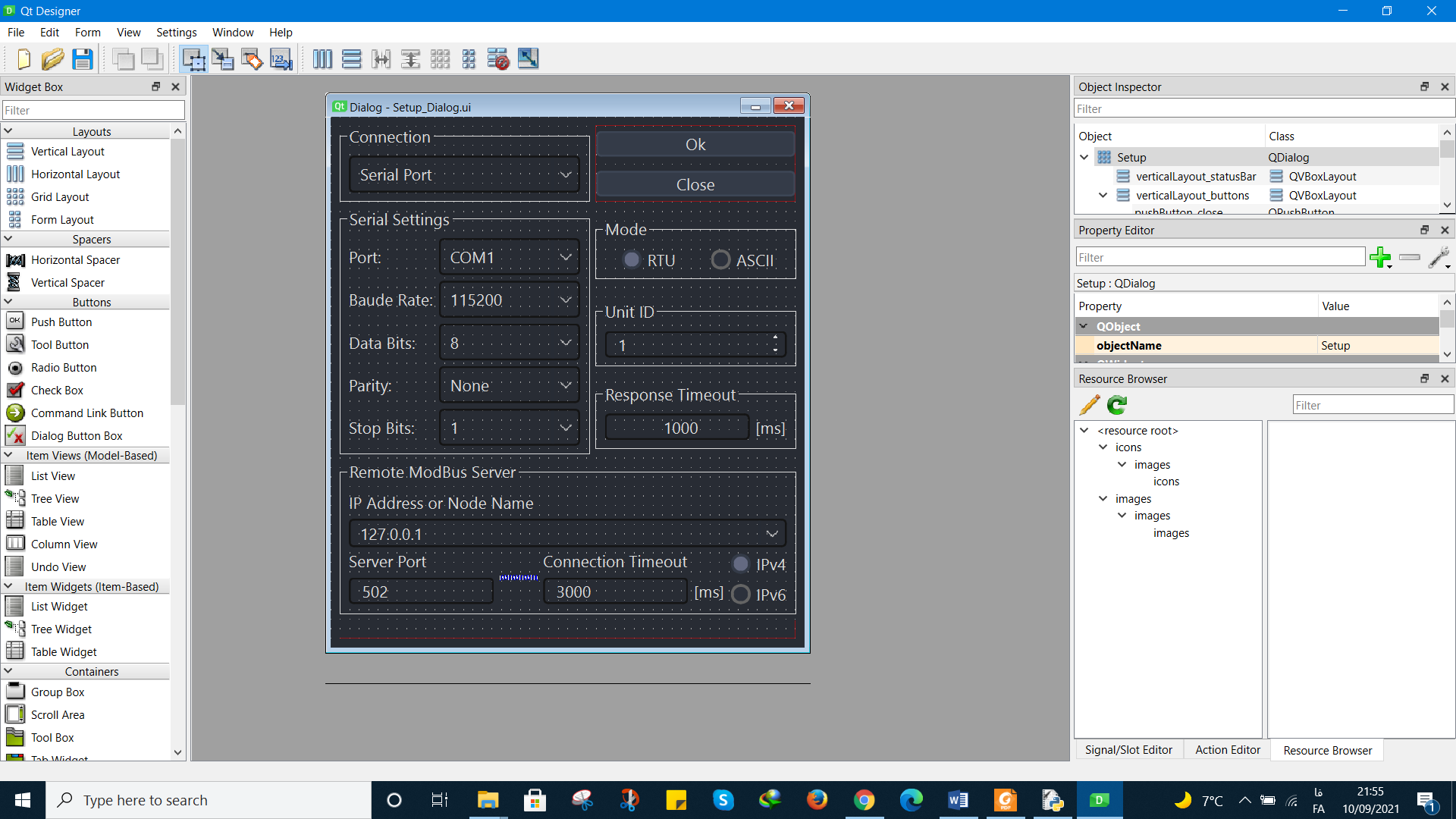
**ابزار**  
PySide شامل ابزارهايي بوده که برنامه‌نويسي را آسان‌تر و سريع‌تر مي‌کند.  
pyside6-uic: معادل ابزار uic مي‌باشد و برگرفته از عبارت User Interface Compiler مي‌باشد. اين ابزار فايل‌هاي رابط گرافيکي ايجاد شده توسط ابزار Qt Designer را به سورس کد پايتون تبديل مي‌کند.  
pyside6-rcc: اين ابزار معادل rcc فريم ورک کيوتي و برگرفته از عبارت Python Resource Compiler v4 مي‌باشد. توسط اين ابزار امکان تبديل فايل‌هاي منبع (resource) به سورس پايتون فراهم مي‌شود. با استفاده از اين ابزار مي‌توان فايل‌هايي چون عکس، آيکون يا هر فايل دلخواه را به سورس پايتون تبديل و در برنامه يا فايل اجرايي مستقل (مثلا exe در ويندوز) دروني‌سازي نمود.

pyside-lupdate: اين ابزار نیز معادل lupdate است و براي ايجاد يا بروزرساني فايل لغات برنامه مورد استفاده قرار مي‌گيرد. با استفاده از اين ابزار مي‌توان فايل ts ايجاد و سپس با استفاده از نرم افزاري چون Qt Linguist آنها را ترجمه و در برنامه استفاده کرد. اين ابزار روند توسعه برنامه‌هاي چند زبانه را بسيار آسان مي‌کند.

**مراحل طراحی برنامه**

برای طراحی یک رابط گرافیکی با استفاده از PyQt، دو روش وجود دارد. روش اول این است که بخش‌های مختلف برنامه را به صورت کد نوشته و اجرا کنیم، اما راه آسان‌تر استفاده از Qt Designer است که قابلیت‌های زیادی در اختیار طراح قرار می‌دهد و می‌توان کلیدها، لیبل‌ها، ادیتورها و به طور کلی ویجت‌ها (Widgets) و اشیا (Objects) را در صفحه مورد نظر قرار داد.

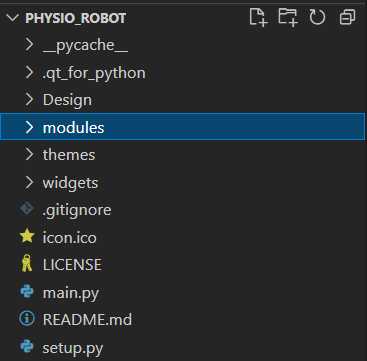
نمای کلی این نرم‌افزار به شکل زیر است:



این نرم‌افزار این قابلیت را به کاربر می‌دهد که برنامه را اجرا کند و فاصله و عملکرد بخش‌های مختلف را در هنگام اجرا مشاهده و سپس اصلاح کند. همانطور که مشاهده می‌شود، دیزاینر بخش‌های مختلفی دارد از جمله منوی ویجت‌ها، اشیای به کار رفته در برنامه، اطلاعات یک شی و همچنین منابع (Resources) استفاده شده در برنامه ازجمله عکس‌ها و آیکون‌ها و هر چیزی که آدرس آن به برنامه داده شود.

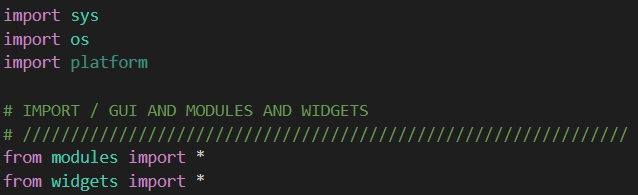
همچنین، با استفاده از امکانات فراهم شده در این برنامه می‌توان خروجی را به صورت کد پایتون یا C++ نیز مشاهده و دریافت کرد و در ادامه از آن در کد اصلی برنامه استفاده کرد.

سپس، می‌توان از کد خروجی تولید شده توسط دیزاینر، درکد اصلی برنامه استفاده کرد (import). مرسوم است که برنامه را در کلاس‌ها و فایل‌های متفاوت پیاده کرد تا دسترسی و تغییر بخش‌های مختلف آن راحت‌تر باشد. برای مثال در این برنامه بخش‌های زیر را داریم:



که در پوشه design فایل‌های طراحی .ui ، در پوشه modules و widgets بخش‌های مختلف کد .py و در فایل main.py کد اصلی قرار می‌گیرد که شامل تمام فایل‌های ذکر شده خواهد بود.

**بخش‌های مختلف کد**



در فایل اصلی کد ابتدا ماژول‌ها، توابع و فایل‌های مورد نیاز را import می‌کنیم. سپس به تعریف کلاس‌های مورد نیاز می‌پردازیم. برای اینکه این فرایند به نتیجه مطلوب‌تری برسد از نوعی دیاگرام به نام UML استفاده می‌شود که در آن کلاس‌ها به همراه توابع و کارکردهاشان، در کنار ارتباط بین کلاس‌ها نمایش داده می‌شود.

**1. Main Window**

در این کلاس فرایندهای اصلی کد آورده شده و کلاس‌های دیگر نیز در این کلاس فراخوانی و اجرا می‌شوند. همچنین صفحه برنامه و بخش‌های مختلف آن در این کلاس اجرا و استفاده می‌شوند. برای ارتباط با دستگاه‌های مختلف، کنترل کردن ارورها، تغییر و ذخیره تنظیمات و ... متدهایی در این کلاس تعریف شده است که دستورات را به کلاس مناسب ارسال می‌کند.

کد این کلاس در صفحات بعد آمده است.

1. **مشکلات تجربه شده.** چه نکاتی چه اشتباهاتی انجام شده بوده که خواننده بایستی نسبت به آنها آگاه باشه که تکرار نشه. چه مشکلات نسبتا قابل توجهی را تجربه کردید که منجر به حل موضوع شد. برای مثال در نرم افزار دستگاه EMG. برای اینکه بتونیم با سرعت دلخواه ثبت بگیریم و نمایش دهیم، کلی مشکلات داشتیم که تیم حل کرد. به نحوی بایستی این مشکلات لیست شوند و راه حل آنها بطور خلاصه در این قسمت بازگو شود. یا برای REAL TIME کردن pc، چه تجربیات اشتباهس داشتیم که برطرف کردیم؟
2. **: جمع‌بندی** این قسمت مهمترین قسمت هست. باید بسیار روان نوشته شده باشد و بگونه ای باشد که اکر خواننده فقط همین قسمت را خواند کل کار را با نتایج متوجه شود. مثلا ابتدا این کار ر ا کردیم و ... در قدم اوا 2.5V ولتاژ دادیم و خطا به 2mm رسید. در قدم بعدی .این کار را کردیم. خلاصه اینکه اگه یکنفر فقط این گزارش را بخونه متوجه خروجی اش کاملا بشه
3. **پیشنهاد برای مرحله بعد:** قدم بعدی چه هست. ممکن هست بگویید تکمیل این گزارش با اضافه کردن این کارها در آینده. اینجا در واقع نقطه ضعف ها بصورت خلاصه مجددا بیان میشه تا شفاف باشه. خلاصه هر چه را که فکر میکنید میتوان در این جا نوشت.
4. **مراجع.** در داخل محتوا زمانی که به یک مرجع، ارجاع میدید – شماره صفحه آن هم بیاد تا بشه پیداش کرد.
5. **پیوست ها**.. در اینجا میتونید تصاویری از یک و یا چند صفحه از کاتولوگ محصولی که استفاده شده، تصویری در اینترنت که کمک کرده. - کل کد کامنت گذاری شده -

دیگه چیز ها:

1. حتما زیر هر تصویری که خودتون کشیدید- از نرم افزاری که خودتون نوشتید – دستگاهی که خودمون ساخته شده زیرش بعد از توصیفش بنویسید به انگلیس  
   Robotics Research at Ferdowsi University of Mashhad - Iran Constructed in Center of
2. حتما تصاویر و جدول ها زیر نویس و شماره داشته باشند و در متن به آنها اشاره شده باشد
3. حتما در صورت نیاز section بندی شود مانند
4. مانند
   1. مانند
   2. مانند
      1. مانند
5. تعداد کل صفحه ها در Footer آپدیت شود
6. در قسمت header عنوان گزارش آپدیت شود
7. به شماره صفحه پایین دقیت کنید. تعداد کل را در نهاین تغییر دهید که شماره صفحه آخر شما 12/50 نباشه

دست همگی برای مستند سازی درد نکنه. این قسمت از کار را معمولا کسی دوست نداره ولی فوقالعاده مهم هست چرا که کمک میکنه اینهمه زحمت و انرژی که کذاشته بشه بطور موثر برای قدم های بعدی استفاده بشه و در نتیجه مرکز تحقیقات را بتونیم به بهترین مرکز تحقیقات رباتیک در دنیا تبدیل کنیم