80％

日本の高校生の実際の勉強時間は自分が勉強をしていたと思っている８０％ほどしかできていないそうです。

これは、一日６時間勉強する人だと、一週間で約９時間、一か月で３８時間もの誤差になります。

実際に私もそうで、参考書は開いているものの、ちゃんと身が入っていなかったり、ぼーっとしたりと無駄な時間を過ごしていたりすることがあります。

そして、これらの課題を解決するために作ったのがStudyWith・実勉強時間測定ソフトです。

そもそもなぜ勉強時間を把握する必要があるのか、それは具体的に数値を出すことで勉強のモチベーションを上げ、さらに勉強目標を立てやすくなり、バランスよく勉強することができるからです。

このソフトの機能は大きく分けて３つあり、

１つ目に　勉強時間計測

２つ目　結果の確認

３つ目　翻訳機能　です

１つ目の計測ですが、活動報告書の通り、今リリースされている多くのソフトの弱みはリアルな勉強時間が計測されない事です、これを解決するために、ペン自体にセンサをつけて傾いている間だけ時間を計測する機構を作りました、そして信号の一連の流れを図にするとこうです。

専用基盤は開発初期から基盤、パーツ共に変更を重ね、最終的には勉強するには支障がきたさない程度まで小型・軽量化する事が出来き精度も向上しました。

また、信号を受信する機構は、発信機から送られた赤外線信号をArduinoを通過することで扱いやすい形に変換し、PythonでAruduinoとシリアル通信するコードを書くことでリアルタイムでの赤外線信号を受信する事が出来るようになりました。

そして、ソフト内では五教科＋その他の教科でそれぞれ勉強時間を計測する事が出来る事が出来、それぞれイラストのボタンを押すことで実際に計測が始まります。

次に計測結果の確認ですが、ホーム画面に戻り左にスクロールすると、一日、一か月に分け、計測された勉強時間がヒストグラムで表示れます。利用者に億劫さを感じさせないよう、シンプルかつ直感的な操作の実現のため、スクロールで画面が切り替わるようにプログラミングし、　それに伴い、GUIライブラリをより自由度の高いKivyに変更しました。

そして、こちらは赤外線信号受信～データ保存までのプログラムを一部抜粋したものです。

２行目で仮想シリアルポートを通じてArduinoと通信し、３行目以降からは信号が入るたびに、データ保存用のCSVファイルの存在確認をし、無い場合はファイルを新規作成、データ保存します。

プログラム後半のCheck、Create、Write\_file関数はそれぞれファイルの存在確認、新規作成、書き込みの機能を持っており、別に定義してあります。

これらの関数はClassをもっと活用することでシンプルで管理しやすく出来たのではないかと思っています。

また、８～１１行目までで専用の赤外線発信機以外から送信された赤外線信号を受信しないように記述しました

これらの一連の処理を一つのthreadとして実行することで、画面描写と並列して実行することが可能になりました。

最後に翻訳機能です。英語の勉強でもそうですが、プログラミングを習得しようと思うと海外のサイトを参考にすることが多くあると思います。そんな時毎回毎回翻訳ソフトやサイトを立ち上げるのは非常に手間がかかります。

そこで、常に起動しているであろうこのソフトのその機能を搭載することで無駄な手間がかからないようにしました。

中でも、画像翻訳ではJPEG,PNGファイルの画像ファイルをアプリのウィンド内にドラッグアンドドロップすることで、その画像のPathを取得、そのPathをもとにその画像をOCR処理します。ここで抽出された文字列はそのまま翻訳ライブラリに引き渡すことでドラッグアンドドロップ以外の一切の確定操作をすることなく翻訳できるようにしました。

もちろん翻訳ソフトを使っている間も計測されます。

最後に改善点です。、

通信手段の汎用性が低かったり、ソフト面ではその日、その月しか結果が確認できないなど、まだまだ改善点は多くありますが、一番大きな課題は、送信機の安定性と耐久性だと思っています。

この発信機の場合センサではなく傾斜スイッチを取り付けているため、チャタリングが発生したり微妙に傾斜があるところにペンを置くだけでも時間が計測されてしまいます。

この傾斜スイッチをジャイロセンサーに変更することで長時間座標の更新がない場合の信号を無視するようにすることができます。

これらの課題を解決した後、将来的にAIを組み込むことで勉強時間とテスト成績などを関連させて分析することで、その人に最適な学習目安時間を提供する事が出来るようになると思います。