Řídící Software

Řídící software pro osazovací automat byla nejtěžší část celého projektu. SW spojuje jednotlivé části popsané v předcházejících kapitolách do jednoho celku.

Jeden z hlavních požadavků na řídící SW byla jeho platformová nezávislost. Tedy možnost spuštění aplikace jak na operačním systému Linux, tak i na Windows. Protože aplikace má grafické rozhraní, zúžil se výběr mnou známých programovacích jazyků na C/C++, Java, Delphi a Python. Byl vybrán právě poslední zmiňovaný Python, jelikož má velice dobrou dokumentaci a nástroje pro tvorbu GUI jsou uživatelsky přívětivé.

Tato kombinace slibovala rychlý prototyping SW a naději na funkční SW. Pro tvorbu GUI padla volba na PyQt.

V aplikaci je kvůli centrování součástek a desek potřebné i vyhodnocování obrazu. Jako základ byla použita hojně používaná knihovna OpenCV. Ta nabízí set základních funkcí pro manipulaci s obrazem. Implementované funkce jako rozostření, hledání hran, kontur a kruhů zjednoduší úlohu rozpoznávání pozice a rotace součástek a hledání centrovacích bodů DPS.

Screenshoty z QTGUI a pár věcí ohledně PyQt

**Data pro osazovací automat**

Pro návrh elektronických systémů se používá software spadající do kategorie EDA – Electronic Design Automation. Je to soubor nástrojů pro tvorbu desek plošných spojů (a integrovaných obvodů). Mezi základní nástroje patří Schématické editory, simulátory obvodů, autoroutery, návrhové prostředí pro tvorbu DPS a CAM procesor.

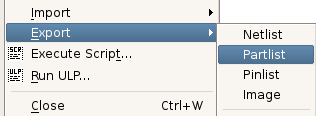
Příkladem EDA softwérů jsou: Altium Designer, KiCad, CadSoft Eagle.

Pro testování byl použit poslední zmiňovaný CadSoft Eagle, který je ve své základní varianě pro nekomerční účely dostupný zdarma. Uvžujme vytvořené schéma a DPS. Pro osazovací automat potřebujeme získat pozici, hodnotu, typ pouzdra a rotaci každé SMD součástky, dále potřebujeme získat pozici centrovacích značek. K tomu se částečně dají použít jendak vestavěné funkce, Eagle ale disponuje i možností použití tzv ULP (User Language Program. ) skriptů. ULP je programovací jazyk postavený na základech C a umožňuje přímé modifikování schématu, DPS a vytváření různých exportů.

<http://www.cadsoftusa.com/downloads/file/ulp720_en.pdf>

Vestavěný export

Jednou z cest jak vyexportovat pozice součástek je File->Export->Partlist



Výsledný export obsahuje všechny použité součástky a exportované pozice jsou v jednotkách mil. Pro osazovací automat je ale potřeba jen SMD součástek a centrovacích bodů. Tento export tedy není příliš vhodný, protože by potřeboval ještě následnou ruční úpravu spočívající minimálně v odmazání všech THD součástek

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | Value | Package | Library | Position (mil) | Orientation |
| C1 | .1uF | C0603 | resistor | (2860 300) | R180 |
| C2 | 18pF | C0603 | rcl | (2075 1405) | R90 |

Ukázka exportu.

Součástí instalace Eagle je i několik již připravených ULP skriptů pro export, například Centroid\_ScreamingCircuits\_smd.ulp

Ten generuje oproti Partlistu strojově čitelnější formát a exportuje jen SMD součástky. Bohužel však chybí typ použitého pouzdra a hodnota součástky.

RefDes,Layer,LocationX,LocationY,Rotation

C1,Top,2.860,0.300,180

C2,Top,2.075,1.405,90

Pro vytvoření exportu se všemi potřebnými hodnotami tak bylo potřeba napsat vlastní ULP skrip.

Ten exportuje středy/origins součástek tak, jak byly vytvořené autorem součástky v knihovnách, dále i geomterické středy součástek. Geomterický střed funguje tak, že se iteruje nad všemi ploškami součástky a hledá se minimum a maximum v obou osách. Jejich rozdíl se vydělí dvěma a najde se skutečný střed součástky. Není to tak střed součástky základě geometrického tvaru pouzdra! Na to je třeba brát později zřetel. Důvod pro export těchto souřadnic je ten, že né všechny součástky v knihovnách se drží zažitého standardu na umisťování středícího bodu do středu pouzdra, případně do levého horního rohu.

Ukázka z ULP skriptu exportující informace o centrovacích bodech DPS

printf("Part name;Package;Value;X origin;Y origin;\n");

printf("%%fiducials\n");

B.elements(E) if (E.populate) {

if (E.package.name == "FIDUCIAL\_1MM")

printf("%s;%s;%s;%.3f;%.3f;\n",

E.name, E.package.name, E.value, u2mm(E.x), u2mm(E.y));

}

printf("%%end\_fiducials\n");

Výsledný export je pak ve formátu

%data

Part name;X center;Y center;X origin;Y origin;Rotation;Value;Package

C1;72.644;7.620;72.644;7.620;180;.1uF;C0603

C2;52.705;35.687;52.705;35.687;90;18pF;C0603

%data\_end

%fiducials

Part name;Package;Value;X origin;Y origin;

U$3;FIDUCIAL\_1MM;FIDUCIAL;7.500;3.000;

U$4;FIDUCIAL\_1MM;FIDUCIAL;7.500;57.000;

U$6;FIDUCIAL\_1MM;FIDUCIAL;92.500;3.000;

%fiducials\_end

skript také exportuje obrázek dané DPS, který se dá po načtení do řídícího SW použít pro simulaci osazování. Celý skript je přiložen v příloze B.

**Firmware**

Firmware slouží jako mezičlánek mezi PC a hardwarem osazovacího automatu. Přijmá příkazy od řídícího SW a ty pak vykonává.

Ověřeným standardem pro instruování CNC strojů jsou tzv G-kódy. Programovací jazyk G, specifikován pod standardem RS274D umožňuje pomocí jednoduchých instrukcí řízení celého stroje. Standard RS274D ale není striktně dodržován a výrobci CNC strojů a řídících kontrolérů si upravují a vytváří vlastní specifickéG-kódy.

Struktura G-kódu je následující: G<číslo> <parametry>.

Pomocí G<číslo> se rozlišuje o jaký příkaz se jedná a <parametry> jsou vstupní parametry příkazu

Jako ukázka poslouží kód na pohyb v osách **G0**, ten bere parametry název a cílovou pozici osy.

G0 X-10.3 Z12

Parametry X-10.3 a Z12 udávají jaké osy se mají kam pohnout. Není však specifikováno, jestli se jedná o absolutní, nebo relativní pohyb. K tomu slouží příkazy G90 (absolutní) a G91(relativní).

Následující posloupnost tedy nastaví pozici stroje do X0, Y10, Z0.4 a poté provede relativní pohyb X5, Z2. Výsledná pozice stroje je tedy X5, Y12, Z0.4

G90

G0 X0 Y10 Z=0.4

G91

G0 X5 Z2

Ne všechny požadované funkce byly ovšem ve firmware dostupné. Chyběla hlavně podpora tlakového senzoru a dále bylo potřea nastavit defaultní konfiguraci pinů při odpojené SD kartě.

V příloze C je možné získat již zkompilovaný firmware ve formátu .bin Z důvodu nadměrné velikosti nejsou zdrojové soubory součástí přílohy.

Aktuální verzi zdrojových kódů modifikovaného firmware je možné získat přes internet za pomocí programu GIT příkazem:

git clone <https://github.com/Hyna/Smoothieware.git>

**Požadavky na naváděcí značky ztv. Fiducials:**

<http://pcbget.ru/Files/Standarts/IPC_7351.pdf>

Naváděcí značky detailně popisuje IPC standard 73 51 konkrétně sekce 3.4.4. Naváděcí značky lze popsat jako geometrické obrazce sloužící k sesouhlasení souřadnicového systému při jednotlivých výrobních operacích.

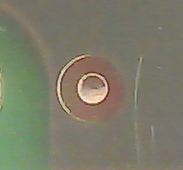
Rozlišují se tři základní druhy značek a to panelové, globál a lokální. Panelové slouží jako reference v případě že panel obsahuje více jednotlivých motivů DPS. Globální pak slouží k lokalizaci jednotlivých komponent na DPS. Poslední kategorií jsou lokální naváděcí značky pro přesné zaměření jednotlivých komponent, zpravidla integrovaných obvodů.

Pro zaměření X a Y pozie DPS a její rotace stačí dvě naváděcí značky. Pro korekci nelineárního zkreslení je zapotřebí minimálně třech značek. Se třemi značkami je tak možné korigovat i chyby v měřítku. Značky by měly být umístěny co nejdál od sebe a tvořit pomyslný trojůhelník.

Obrázek jednotlivých centrovacích značek.

Optimální vzhled centrovací značky by dle standardu by měla mít formu kruhu o průměru 1mm tvořeného mědí. Připouští se povrchová úprava a to ideálně OSP. Okolo kruhové oblasti tvořené mědí s poloměrem **r** je další kruhová plocha o poloměru **2\*r** a to bez mědi a nepájivé masky viz následující obrázek:

obrázek centrovacích značek + naznačení poloměrů



V případě vícevrstvých DPS je vyžadováno pod všemi centrovacími značkami stejné pozadí. Tzn se nedoporučuje vést ve vrstvě přímo pod centrovacími značkami vodivé cesty.

K detekci centovacích značek je použit obraz z horní kamery. Vyhledat přesnou pozici centrovací značky lze pomocí vyhodnocování obrazu a to dvěma způsoby. Buď za pomocí referenční šablony a nebo pomocí detekce určitých rysů v obraze- zde kruhů.

Metoda s použitím naučené šablony dosahuhje přesnějších výsledků, avšak není příliš univerzální. Při změně velikosti centrovací značky a nebo při změně barvy nepájivé masky se její přesnost snižuje.

Oproti tomu použití detekce rysů v obraze si dokáže spolehlivě poradit i s různou velikostí centrovacích značek. V našem případě se hledají kruhy a to za pomocí Houghovy transformace. Právě tato univerzální metoda byla použita do řídícího systému.

<http://en.wikipedia.org/wiki/Hough_transform>

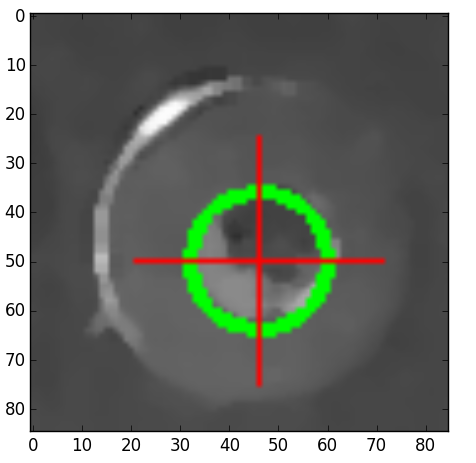
**Vliv osvětlení na detekci centrovacích značek.**

V průběhu testování se ukázalo, že světelné podmínky mají velký vliv na přesnost a spolehlivost detekce. Ideálních podmínek bylo dosaženo při eliminaci všech vnějších světelných zdrojů a přisvětlení pomocí LED diod.

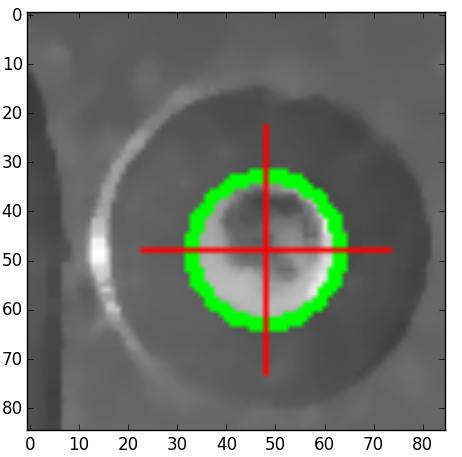
Napsat komentář k přesnostem.

Ukázka vlvivu osvětlení na detekci:

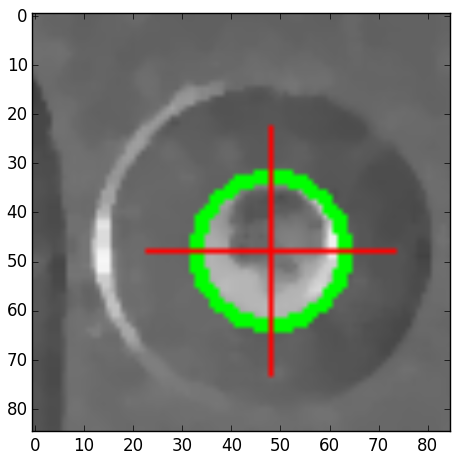
fiduc\_denni\_crop.png **46, 44**



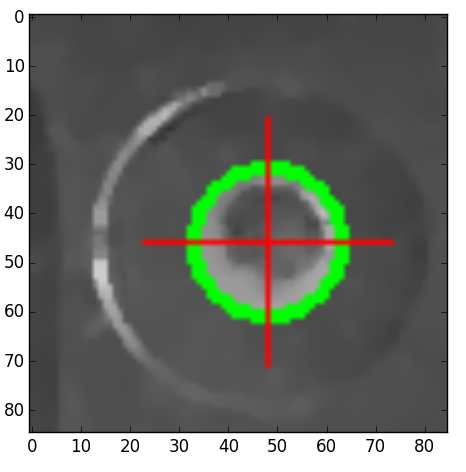
fiduc\_tma\_prisviceno\_crop.png **48, 50,**



fiduc\_svetlo\_prisviceno\_crop2.png **50, 46**



fiduc\_svetlostrop\_crop.png **46, 44**





Ukázat testy ze skriptu fiducials-loop.py

<http://wikieducator.org/OpenOffice/Calc_3/Histogram>

Měření rychlosti IPC-9850

<http://jpk.sdp.edu.cn/dqx/xxwz/5xxzy/hybz3-IPC9850.PDF>

zajimavé čtení

<http://www.academia.edu/1603233/Understanding_IPC-9850-To_avoid_misunderstandings_purchasers_should_compare_the_accuracy_and_DPMO_specifications_at_the_IPC_reference_speed>\_