Slide 1

Vazena porota mili hostia, moje meno je patrik pekarcik a v nasledujucich minutach vam predstavim opensourcovy projekt ACP a ukázem na com som v projekte pracoval.

Slide 2

Pojmy Arduino a internet veci idu spolu. Internet veci najlepsie znazornuje tato schema, kde sipka znazornuje primarny smer toku dat pre buduce rozhodvanie. Vacsina dostupnych rieseni robi vsak to ze lokalne spracovanie je len preposielanie cez internet na cloud. Firmy venujuce sa IoT ako napr. aj cisco vo svojich clankoch zaviedli pojmy fog computing resp. edge computing. Jeden z hlavných messagov ktore z toho vyplyvaju je rozsierenie pouzivania lokalneho processingu. Lokalny processing vsak znamena komplexnejsi kod na mikrokontroleroch.

Slide 3

Komplexny kod vsak nie je ziaden problem. Mame obrovsku podporu integrovanymi vyvojovymi prostrediami a generatormi kodu. Ci uz to su konfiguracie (napr. ORM modelov), windows forms, aktivity alebo aj workflow generatory.

Slide 4

Spomenul som generovanie kodu vdaka ktoremu mame jednoduchy vyvoj, kratsi cas a znovupouzitelne komponenty. Na druhej strane vsak ziskavame vseobecny a neoptimalizovany kod, ktory nie je jednoduche ladit. Vo svete 8GB Ram a 2 Ghz procesorov to vsak nie je ziadne obmedzenie. Problem nastava ked si uvedomime nizke parametre MCU. Tu nam asi vela moznosti neostava ako stare dobre C++.

Slide 5

Rozhodli sme neinvazinve vytvorit generator c++ kodu ktory sa na projekt pozera inak ako sme zvyknutý. Projekt vidí len komponenty, cize display, senzor vzdialenosti, MCU a tieto prepája. Device = Typ komponentu = 3jica mnozín Vlastnosti, Udalosti, Nazvy metod v c++. ACP projekt je nasledne spojenie konfiguracie a c++ obslusneho zdrojoveho kodu.

Slide 6

Projekt sa sklada z troch hlavnych casti, Generatora IDE a kniznice prepripravenych typov komponentov. Generator je samostatne aby nebolo nutne vyuzitie IDE a dodava to moznosti vyuzitia v continuous integration pripadne aj deployment. IDE si o chvilu ukazeme a kniznica typov komponentov je ako maven repozitar pre nas projekt. Ludia tam mozu davat nove typy a iny ich budu pouzivat vo svojich projektoch.

Slide 7

Pozrime sa ako IDE vyzera a pracuje sa s nim. Z toolboxu si vyberieme typ, kliknutim na neho nastavime vlastnosti a udalosti. A v zdrojovom kode implementujeme co sa stane. Ako vidime mozeme vyuzit automaticke dokoncovanie a generovanie metod.

Slide 8

Videli sme ako sa to pouziva. Podme sa pozriet pod kapotu projektu a povedat ake problemy sme museli vyriesit. Azda prvym problemom bolo porozumiet zdrojovemu kodu. Neda sa to robit obycajnym skenerom, pretoze by sme sa velmi jednoducho dostali neziaduci vystup napr. zakomentovany. Potrebovali sme riesit problemy ake sa riesia pri tvorbe kompilátorov a vytvorit tak abstraktný syntaktický strom na zaklade specifikacie jazyka c++.

Slide 9

Je to nieco take, prechod kodom je to iste ako ten strom DFS-kom. Na vygenerovanie stromu sme potrebovali vytvorit gramatiku jazyka, pomocou terminalov, neterminalov a gramatickych pravidiel. Na porozumenie znakov sme naviac potrebovali lexikanu analyzu pomocou ktorej pomenovamane slova ako WHILE IF alebo aj regularnymi vyrazmi predpisujeme ako mozu vyzerat nazvy funkcii tried a pod. Taktiez ziskavame validaciu spravnosti zdrojoveho kodu, kedze gramatikou prejde len spravne napisany kod. Ak je niekde chyba tak vieme programatora upozornit vyznacenim presneho miesta s chybou.

Slide 10

Pozor. z ast sme nedostali len strom! Dostali sme metadata o zdrojovom kode, kde co v nom je ake su funkcie, ake maju parametre, premenne importa a mnoho mnoho dalsieho.

Slide 11

Pomocou zozbieranych metadat uz vieme robit tieto overenia projektu, a ponukat existujuce metody ako udalosti.

Slide 12

Dalsim problemom ktory sme riesili bola aj medzi procesova komunikacia. Kedze pre kompilaciu a posielanie do arduina vyuzivame Arduino kompilator potrebujeme spustit tento proces z javy. Proces je spusteny samozrejme v samostatnom vlakne ale ako zabezpecit aby ide dokazalo efektivne spracovat vystup z kompilatora ak vystup prichadza v 2 streamoch? A to nerohovim este o tom aby sa IDE mohlo stale pouzivat.

Slide 13

V praci som sa venoval aj implementacii typu komponentu do kniznice, za zmienku stoji tento sluziaci pre radiovu komunikaciu frekvenciou 433 mhz. Frekvencia je zaujimava najmä svojim pouzivanim v garazovych branach alebo zasukach na dialkove a inych dialkovo ovladanych zariadeni v domacnosti. Pomocou tohto modulu z coho kolvek vieme urobit smart zariadenie.

Slide 14

Dakujem za pozornost a tesim sa na vase otazky