

Realizacja systemu akwizycji danych z urządzeń przemysłowych

Autor:

Imię i nazwisko: Adam Walczak
Email: walec51@o2.pl
Uczelnia: Politechnika Poznańska
Wydział Elektryczny
Informatyka
Numer indeksu: 70979

Wersja systemu:

Numer wersji: 1.05
Data ukończenia: 04.04.2007

Spis treści

1.Wstęp.....	1
2.Specyfikacja wymagań.....	2
3.Wykorzystane narzędzia.....	4
4.Realizacja systemu.....	8
4.1.Struktura systemu.....	8
4.2.Dokumentacje układów.....	11
4.3.Dokumentacje programów grupy roboczej.....	13
4.4.Dokumentacje programów grupy zarządzającej.....	18
4.5.Baza danych SCdb.....	26
4.6.Kody źródłowe.....	30
4.7.Formaty plików.....	31
5.Wdrożenie systemu.....	34
5.1.Wdrożenie grupy roboczej.....	34
5.2.Wdrożenie grupy zarządzającej.....	35
6.Podsumowanie.....	38
Zawartość płyty CD.....	39
Literatura.....	41

1. Wstęp

Wprowadzenie

Niniejsza praca inżynierska opisuje realizację systemu akwizycji danych z maszyn przemysłowych wykonanego przez autora tej pracy dla firmy Walczak (www.walczakparkiety.pl). Wykonany system nosi nazwę Selection Check, w skrócie SC.

Cel pracy

Celem tej pracy jest opisanie realizacji oraz budowy systemu Selection Check.

Zakres pracy

Zakres pracy obejmuje specyfikację wymagań (roz. [2.Specyfikacja wymagań](#)), w której opisane są motywacje i założenia na bazie, których powstał system Selection Check. Dalszą część pracy obejmuje opis narzędzi i technologii (roz. [3.Wykorzystane narzędzia](#)) wykorzystanych do realizacji systemu oraz uzasadnienie ich wyboru. Główną częścią pracy jest opis realizacji systemu (roz. [4.Realizacja systemu](#)), w której nakreślona jest ogólna budowa system, jaki i szczegóły implementacyjne. Na końcu opisane zostały także ogólne instrukcje dotyczące wdrażania systemu (roz. [5.Wdrożenie systemu](#)), na przykładzie wdrożenia w zakładzie firmy Walczak.

2. Specyfikacja wymagań

Podstawową motywacją do powstania systemu jest potrzeba monitorowania wydajności pracowników pracujących na skracarkach w firmie Walczak (www.walczakparkiety.pl). W tym celu należy opracować system, którego zadaniem będzie pobieranie z skracarek informacji na temat wyciętych fryzów oraz generowanie na podstawie tych danych raportów na temat wydajności poszczególnych pracowników. Sam system od strony informatycznej powinien być stosunkowo prosty i bardziej służyć do prototypownia rozwiązań informatycznych na hali produkcyjnej, a niżeli obsługiwać je w pełnym zakresie.

Opis pracy na skracarkach

Na skracarkach w tartaku wykonywana jest tzw. manipulacja wzdłużna surowcem. Otóż na stanowisko dostarczane są łaty drewna, które należy podzielić wzdłuż na fryzy o długościach określonych przez zakład. Przy podziale każdego fryzu należy uwzględnić rozkład jakości drewna, określone przez zakład długości fryzów i klasy sortowania. Optymalnie z każdej łaty należy uzyskać jak najdłuższe fryzy o jak najwyższej jakości. Statystycznie sama średnia długość oraz ilość wyciętych przez pracownika fryzów jest w stanie określić jego wydajność.

Pobieranie danych ze skracarek

Rejestrowanie danych z skracarek nie może w żaden sposób utrudniać pracy pracownikom. Do rejestrowania długości zaleca się wykorzystanie ograniczników wykorzystywanych przez pracowników przy wymierzaniu fryzów. W każdej ze zaadaptowanych skracarek powinno być rejestrowane zdarzenie w postaci wycięcia poprzez pracownika fryzu, a wraz z nim następujące dane:

- czas wycięcia fryzu z dokładnością do jednej sekundy,
- stan czujników rejestrujących długość fryzu,
- identyfikator skracarki.

Serie wyżej wymienionych danych powinny być przechowywane na hali produkcyjnej z możliwością ich transferu do bazy danych znajdującej się w biurze. Transfer danych będzie zazwyczaj wykonywany raz w tygodniu.

Kontrola wydajności pracowników

Z danych uzyskanych w bazie danych system powinien generować raporty na temat wydajności poszczególnych pracowników. Do realizacji tego celu do systemu będzie wprowadzany harmonogram pracy pracowników na skracarkach.

System powinien mieć możliwość generowania następujących raportów:

- ilość wyciętych fryzów o danych długościach przez wszystkich pracowników w danym miesiącu,
- ilość wyciętych fryzów o danych długościach przez wszystkich pracowników w danym dniu,
- ilość wyciętych fryzów o danych długościach na danej skracarce w danym przedziale czasu.

Koszt wdrożenia

Koszt wdrażania systemu powinien być utrzymany na niskim poziomie aby mógł być wykorzystywany do prototypowania rozwiązań informatycznych. Mimo to użyteczność systemu powinna być zachowana w stopniu pozwalającym zrealizować jego podstawowe założenia.

Modularna budowa

System powinien być wykonany w sposób modularny aby w przyszłości istniała możliwość przystosowania go do pracy z innymi maszynami lub też do analizowania danych pod innym kontem niż wydajność pracowników. Dlatego też część systemu odpowiedzialna za pobieranie danych powinna być oddzielona od części analizującej te dane. W tym celu należy także ustandaryzować formę transferu danych z jednej części do drugiej.

3. Wykorzystane narzędzia

System CMake

CMake (Cross-platform Make) to wieloplatformowe narzędzie do automatycznej kompilacji. Jest podobne do Uniksowego programu make, w którym proces kompilacji jest kontrolowany przez specjalne pliki (Makefile). W przypadku CMake są to pliki o nazwie CMakeLists.txt. CMake, w odróżnieniu od make nie kompiluje kodu samodzielnie, tworzy tylko pliki z regułami kompilacji (np. plik Makefile dla programu make lub plik projektu dla programu Microsoft Visual Studio). Takie podejście pozwala zachować niezależność od platformy, na której kod będzie kompilowany. CMake potrafi kompilować biblioteki (zarówno statyczne jak i dynamiczne) i pliki wykonywalne w dowolnej kolejności. Ponadto pozwala przeprowadzić wiele procesów kompilacji na podstawie tych samych źródeł [4].

W systemie Selection Check jest wykorzystywany przy kompilacji całego oprogramowania.

Kompilatory GNU GCC i MinGW

GCC (ang. GNU Compiler Collection) to zestaw kompilatorów do różnych języków programowania rozwijany w ramach projektu GNU i udostępniany na licencji GPL oraz LGPL. GCC jest podstawowym kompilatorem w systemach uniksopodobnych przy czym szczególnie ważną rolę odgrywa w procesie budowy jądra Linuksa. Początkowo skrótowiec GCC oznaczał GNU C Compiler, ponieważ był to kompilator wyłącznie do języka C [2].

MinGW jest portem części narzędzi programistycznych GNU na platformę MS Windows. W skład pakietu wchodzi kompilatory GCC, zestaw programów binutils (zawierający m.in. kompilator zasobów windowsowych windres) oraz zestaw plików nagłówkowych i bibliotek charakterystycznych dla platformy Windows.

W systemie Selection Check wykorzystane zostały kompilatory GCC dla języków C i C++. Zeswata MinGW został wykorzystany przy kompilowaniu oprogramowania grupy zarządzającej systemem (patrz roz. [4.4.Dokumentacje programów grupy zarządzającej](#)) dla systemów MS Windows.

Platforma Qt firmy Trolltech

Qt to zestaw przenośnych bibliotek i narzędzi programistycznych dedykowanych dla języka C++. Ich podstawowym składnikiem są klasy służące do budowy graficznego interfejsu programów komputerowych, aczkolwiek począwszy od wersji 4.0 Qt zawiera też narzędzia do tworzenia programów konsolowych i serwerów. Właścicielem Qt jest norweska firma Trolltech. Qt jest dostępna dla wielu platform: X11 (GNU/Linux, *BSD, Solaris...), Windows, Mac OS X oraz dla tzw. urządzeń wbudowanych (ang. embedded)

opartych na Linuksie. Qt jest podstawą dla m.in. uniksowego środowiska graficznego KDE, przeglądarki internetowej Opera oraz tak popularnych programów, jak Google Earth i Skype. Biblioteki Qt zostały napisane w języku C++ i charakteryzują się w pełni obiektową architekturą. Zawierają wiele tzw. nowoczesnych technologii programowania graficznego interfejsu użytkownika, istniejących wcześniej jedynie w Tk: mechanizm sygnałów i slotów, automatyczne rozmieszczanie widżetów oraz zhierarchizowany system obsługi zdarzeń. Biblioteki Qt, oprócz obsługi interfejsu użytkownika, zawierają także niezależne od platformy systemowej moduły obsługi procesów, plików, sieci, grafiki trójwymiarowej (OpenGL), baz danych (SQL), języka XML, lokalizacji, wielowątkowości, zaawansowanej obsługi napisów oraz wtyczek [3]. Zawierają także własne, niezależne od STL szablonów klas pojemników.

W skład Qt wchodzi wiele specjalistycznych narzędzi programistycznych. Są to m.in. [3]:

- moc (Meta Object Compiler) - specjalny preprocesor, który na podstawie plików nagłówkowych (*.h) generuje dodatkowe pliki źródłowe (*.cpp).
- uic (User Interface Compiler) - kompilator plików *.ui zwykle generowanych za pośrednictwem programu Qt Designer.
- qmake - program do zarządzania procesem kompilacji; jego głównym zadaniem jest utworzenie, a później aktualizacja pliku Makefile na podstawie prostego opisu zawartego w definicji projektu (*.pro).
- Qt Designer - aplikacja graficzna do definiowania graficznego interfejsu użytkownika (okien dialogowych itp.).
- Qt Linguist - aplikacja wspomagająca tłumaczenie programu na różne języki.
- Qt Assistant - aplikacja zawierająca rozbudowany system pomocy dla programistów.

Od opublikowanej w 2000 roku wersji 2.2, Qt dla platform GNU/Linux/Unix jest rozpowszechniana także na licencji GPL. Obecnie dostępnych jest kilka rodzajów licencji - komercyjna, GPL, QPL oraz edukacyjna. Od 2004 roku licencja GPL obejmuje także platformę Mac OS X, a od 2005 r. (od wersji 4.0.0) - także Microsoft Windows.

W systemie Selection Check platforma Qt jest wykorzystana przy tworzeniu front-end'a dla programów konsolowych grupy zarządzającej (patrz roz. [4.4.Dokumentacje programów grupy zarządzającej](#)).

System dokumentacji Doxygen

Doxygen to generator dokumentacji dla języków C++, C, Java, Objective-C, Python, IDL i do pewnego stopnia dla PHP, C#, D oraz ActionScript. Obsługuje następujące formaty wyjściowe: HTML, CHM, RTF, PDF, LaTeX, PostScript oraz strony man.

W systemie Selection Check Doxygen jest wykorzystane przy dokumentowaniu wszystkich kodów źródłowych.

Powłoka Bash

Bash to powłoka systemowa UNIX napisana dla projektu GNU. Nazwa jest akronimem od Bourne-Again Shell (ang. znowu powłoka Bourne'a). Jest to jedna z najpopularniejszych powłok systemów uniksowych. Wywodzi się od powłoki Bourne'a sh, która była jedną z pierwszych i najważniejszych powłok systemu UNIX, i zawiera pomysły zawarte w powłokach Korna i csh. Pierwotna powłoka Bourne'a była tworzona przez Stephena Bourne'a. Bash był pisany głównie przez Briana Foksa i Cheta Rameya w roku 1987.

Bash jest domyślną powłoką w większości dystrybucji systemu GNU/Linux oraz w systemie Mac OS X w wersji 10.4 Tiger. Istnieją także wersje dla większości systemów uniksowych. Bash jest także domyślną powłoką w środowisku Cygwin dla systemów Win32. Program jest rozprowadzany na licencji GPL.

W systemie Selection Check powłoka Bash jest została wykorzystana przy tworzeniu i wdrażaniu oprogramowania. Skrypty Bash są także wykorzystywane do konfiguracji transferu danych wykonywanego przez program SCwl2dir (patrz roz. [4.3.2.SCwl2dir](#)) oraz konfiguracji czynności wykonywanych w systemie po zakończeniu pracy programu SCrunner (patrz roz. [4.3.1.SCrunner](#)).

Baza danych MySQL

MySQL to, obok PostgreSQL oraz Firebird, wolnodostępny system zarządzania relacyjnymi bazami danych. MySQL tworzony jest przez szwedzką firmę MySQL AB. MySQL był pisany raczej z myślą o szybkości, niż kompatybilności ze standardem SQL – przez dłuższy czas MySQL nie obsługiwał nawet transakcji, co było zresztą głównym argumentem przeciwników tego silnika bazodanowego. MySQL wspiera większą część obecnego standardu ANSI/ISO SQL (tj. SQL:2003) [5]. Wprowadza również swoje rozszerzenia i nowe elementy języka

W wersji 5 dodano m.in. [5]:

- procedury składowane (ang. stored procedures) – obecne w wersji 5.1 beta,
- kursory – obecne w wersji 5.1 beta,
- wyzwalacze (ang. triggers) – obecne w wersji 5.0.2,
- perspektywy (ang. views),

co bardzo zbliża najnowsze wersje MySQL do PostgreSQL pod względem funkcjonalności.

MySQL cieszy się natomiast opinią jednego z szybszych serwerów bazodanowych, dzięki czemu znakomicie nadaje się jako serwer dla często odwiedzanych witryn WWW. MySQL jest dostępny na licencji GPL, jednakże można nabyć również wersje licencjonowane komercyjnie. Biblioteki klienckie są również dostępne na licencji GPL lub komercyjnej. Istnieje wymóg nabycia licencji komercyjnej jeśli zamierzamy dystrybuować aplikację komercyjną (niezgodną z licencją GPL), korzystającą z MySQL.

W systemie Selection Check baza danych MySQL 5 jest wykorzystywana do przechowywania danych w grupie zarządzającej systemu (patrz roz. [4.5.Baza danych SCdb](#)).

Środowisko CAD Eagle

Eagle to środowisko CAD do wspomagania projektowania obwodów elektrycznych niemieckiej firmy CadSoft. Zawiera edytor schematów i płytek drukowanych. Dostępny dla systemów operacyjnych: Windows, Linux, MacOS X.

W systemie Selection Check środowisko Eagle zostało wykorzystane do zaprojektowania układu SCslave (patrz roz. [4.2.1.SCslave](#)).

4. Realizacja systemu

4.1. Struktura systemu

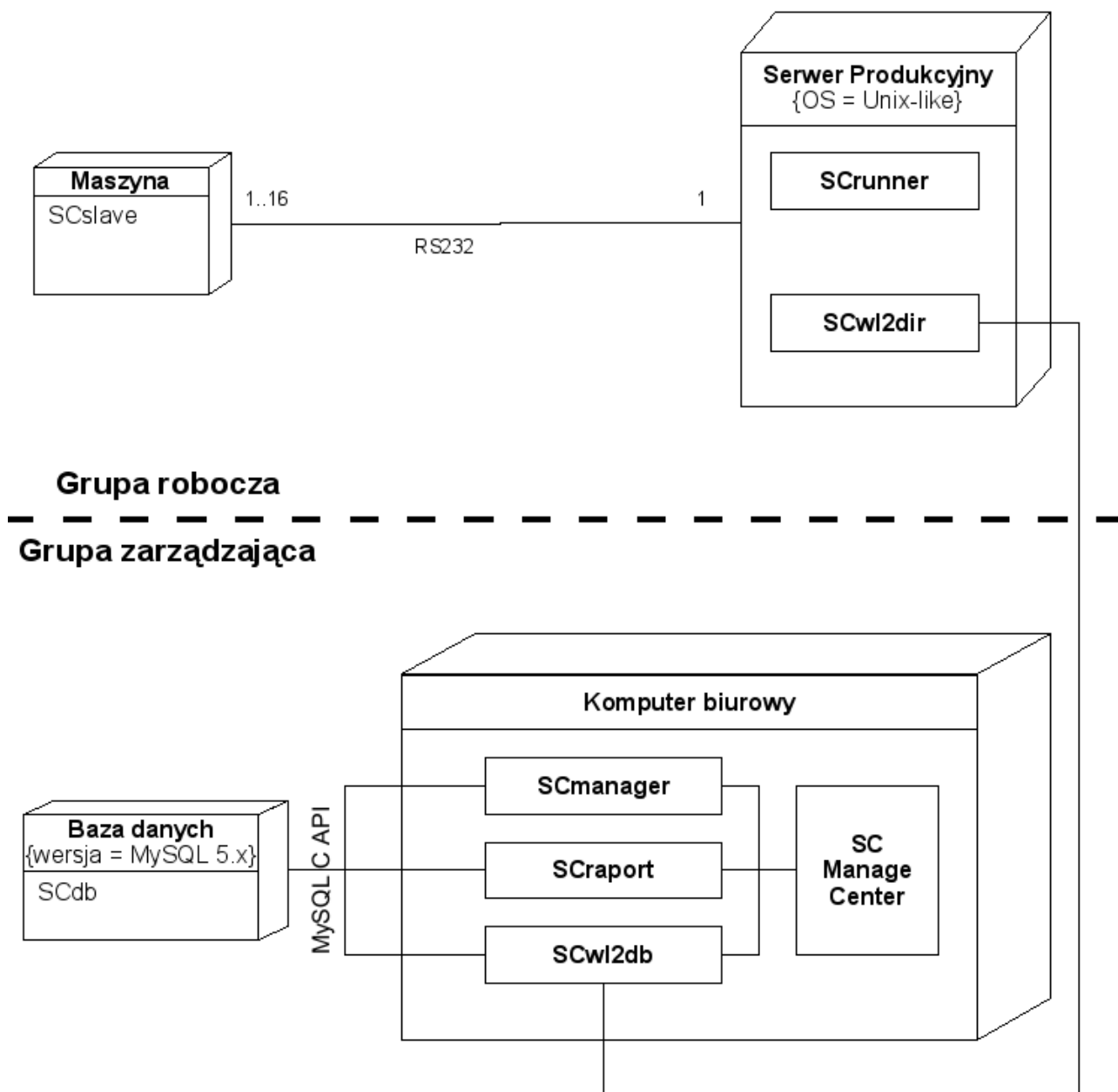
4.1.1. Elementy systemu.

Elementy systemu SC dzielą się na dwie grupy:

- Grupa Robocza (`work_grp`) - są to elementy odpowiedzialne za pobieranie danych z zaadoptowanych urządzeń. Głównymi elementami tej grupy są:
 - moduł `SCslave` – rejestr przechwytyjący dane z zaadoptowanych urządzeń,
 - program `SCrunner` – sterownik w. w. układów oraz logger pobranych z nich danych,
 - program `SCwl2dir` – podprogram sterownika `SCrunner` odpowiedzialny za przesłanie danych pobranych z modułów `SCslave` do katalogu transferowego,
 - program `SCrunner_tui` – tekstowy front-end dla programu `SCrunner`, oparty o `ncurses`,
- Grupa Zarządzająca (`manage_grp`) - są to elementy odpowiedzialne za archiwizację danych pobranych z modułów `SCslave` oraz generowanie na podstawie tych danych raportów. Do tej grupy należą:
 - baza danych `SCdb`,
 - zestaw programów klienckich dla bazy danych `SCdb`:
 - `SCwl2db` – program, który odbiera dane z katalogu transferowego i przekazuje je do bazy danych `SCdb`,
 - `SCmanager` – zarządzanie danymi znajdującymi się wewnątrz bazy danych `SCdb`,
 - `SCraport` – generator raportów na podstawie danych z `SCdb`.
 - program `SC Manage Center` – nakładka graficzna na programy klienckie bazy danych `SCdb`, oparty o platformę `Qt`

4.1.2. Rozmieszczenie elementów.

W tym dziale zostanie przedstawione rozłożenie poszczególnych elementów wdrożonego systemu SC. Sposób w jaki przepływają dane pomiędzy elementami systemu wyjaśniony jest w następnym rozdziale. Poniższy rysunek 4.1 przedstawia rozmieszczenie elementów systemu SC.

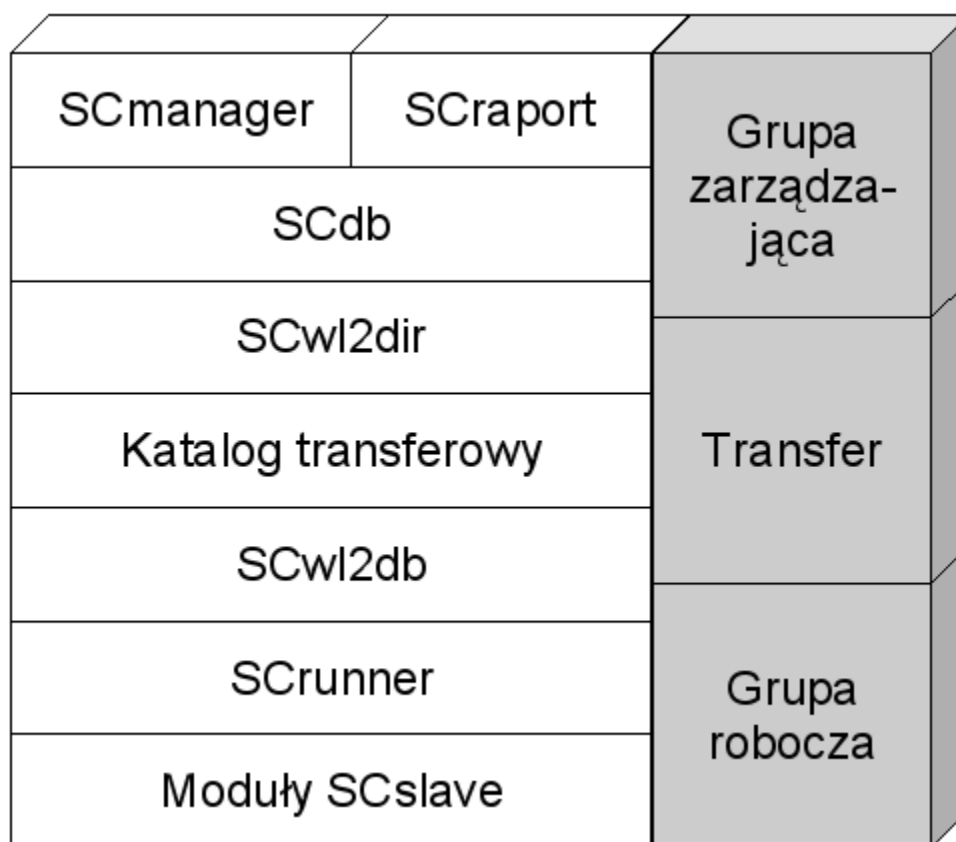


Rys. 4.1. Diagram wdrożenia.

Serwer produkcyjny znajduje się na hali produkcyjnej wraz z zaadaptowanymi urządzeniami. Moduły SCslave poszczególnych zaadaptowanych urządzeń są podłączone do serwera produkcyjnego poprzez łączy RS232. Sterowaniem procesu gromadzenia danych z modułów SCslave zajmuje się program SCrunner. Do analizy danych zebranych z układów SCslave wyznaczony jest komputer biurowy z bazą danych MySQL 5.x oraz z zainstalowanymi programami grupy zarządzającej systemu SC. Medium transferu danych pomiędzy serwerem produkcyjnym a komputerem biurowym jest stosunkowo dowolne.

4.1.3. Przepływ danych.

System SC można zdekomponować do postaci komunikujących się warstw. Poszczególne warstwy systemu przedstawia rysunek 4.2.



Rys. 4.2. Warstwy systemu SC.

Sterownik SCrunner cyklicznie pobiera dane z modułów SCslave i archiwizuje je. Na życzenie użytkownika lub sygnał od innego programu następuje transfer tych danych. Warstwa transferu składa się z programu SCwl2dir, katalogu transferowego oraz programu SCwl2db. Program SCwl2dir przekazuje dane nagromadzone od czasu ostatniego transferu przez program SCrunner do katalogu transferowego. Jak wspomniano w poprzednim rozdziale medium transferu jest stosunkowo dowolne. Otóż katalog transferowy można zdefiniować jako katalog znajdujący się na pendrive'ie, czy też jako katalog sieciowy NFS lub Samba. Na życzenie użytkownika program SCwl2db odbiera dane z katalogu transferowego i umieszcza je w bazie danych SCdb. Operowaniem na bazie danych SCdb zajmują się programy konsolowe: SCmanager, SCraport i SCwl2db. Programy te do łączenia się z bazą danych wykorzystują standardową bibliotekę MySQL dla języka C. Program SCmanager służy do zarządzania danymi zebranymi w bazie danych, natomiast SCraport generuje na ich podstawie raporty. Na rysunku 4.2 nie przedstawiono programów SC Manage Center i SCrunner_tui gdyż są to jedynie front-end'y na wymienione wyżej programy konsolowe.

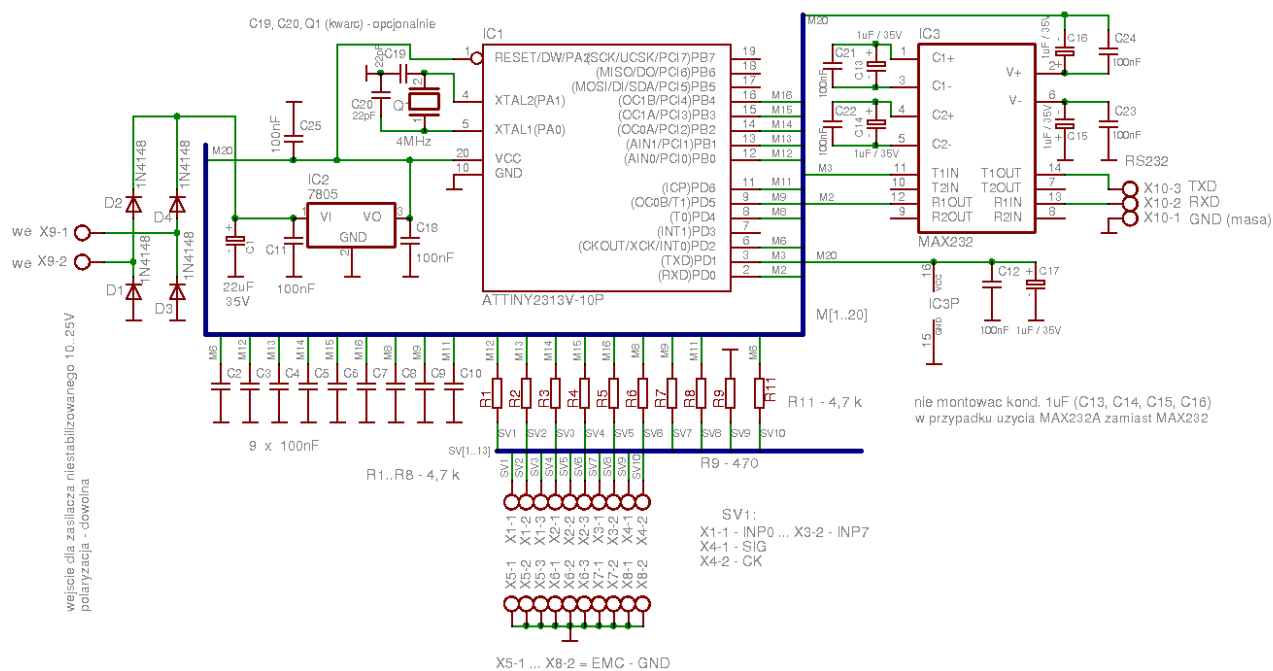
4.2. Dokumentacje układów

Schematy logiczne jak i schematy płytek zostały opracowane w programie Eagle 4. Pliki projektów Eagle znajdują się w katalogu ./hardware/. Obecnie w tym dziale znajduje się jedynie moduł SCslave. Planowane jest także zaprojektowanie prostego panelu sterowania do komunikacji z sterownikiem SCrunner.

4.2.1. SCslave.

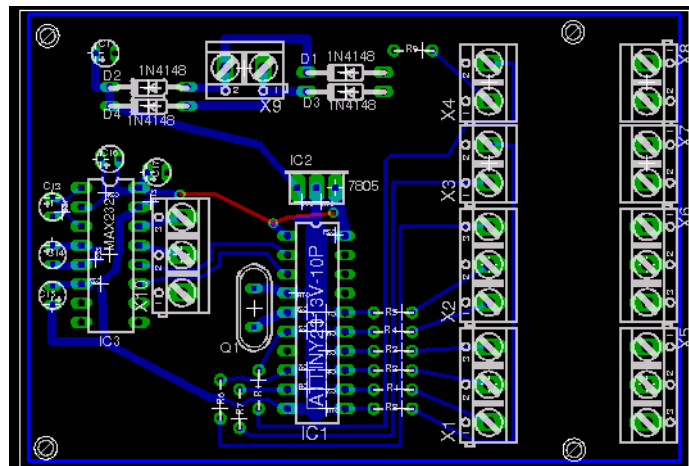
Rejestr pobierający dane z zaadoptowanej maszyny. Pliki projektu Eagle (scslave.sch i scslave.brd) znajdują się w katalogu ./hardware/scslave. W tym katalogu znajduje się także plik źródłowy programu sterującego, rezydującego na mikrokontrolerze modułu. Moduł został zaprojektowany i wykonany przez Pawła Potiomkina na zlecenie autora systemu.

Schemat logiczny.



Rys. 4.3. Schemat logiczny.

Schemat PCB.



Rys. 4.4. Schemat PCB.

Parametry.

Parametry zasilacza:

- Napięcie: 10-25 V, stałe,
- Polaryzacja: dowolna,
- stabilny lub obciążony niestabilny.

Parametry komunikacji poprzez RS232 są następujące:

- szybkość: 4800 bps,

Paczka danych (tabela 4.4):

bit startu	8 bitów danych (I 0-7)	bit parzystości (even)	bit stopu
------------	---------------------------	---------------------------	-----------

Tab. 4.4. Paczka danych.

Komunikacja.

Moduł SCslave jest połączony do serwera produkcyjnego za pomocą RS232. Komunikacja z sterownikiem SCrunner przebiega według następującego schematu:

1. SCrunner wysyła do modułu SCslave znak 'c'.
2. SCslave odpowiada bajtem 0xFF jeżeli posiada w rejestrze nowe dane. W przeciwnym wypadku odpowiada bajtem 0x00.
3. Jeżeli SCslave posiada nowe dane to sterownik wysyła do modułu znak 'd'.
4. SCslave odpowiada bajtem danych pochodzącymi z rejestru.

4.3. Dokumentacje programów grupy roboczej

Oprogramowanie grupy roboczej składa się z programów rezydujących na serwerze produkcyjnym. Wszystkie z nich działają na systemach z rodziny Linux.

4.3.1. SCrunner.

Sterownik obsługujący układy systemu SC. Sposób w jaki sterownik komunikuje się z układami SC jest opisany w dokumentacjach poszczególnych układów (patrz roz. [4.2.Dokumentacje układów](#))

Cykl pracy.

1. Rozruch programu.
2. Półki nie zażądano wyłączenia programu lub nie wystąpił błąd krytyczny wykonuj:
 - 2.1. Odczekaj czas ustalony w cycle delay.
 - 2.2. Dla wszystkich ustalonych układów typu SCslave powtarzaj.
 - a) Sprawdź dane na module SCslave.
 - b) Jeżeli komunikacja z modulem SCslave przebiegła poprawnie oraz moduł przesłał nowe dane to zaloguj te dane.
 - c) Jeżeli przekroczona została maksymalna ilość nieudanych prób komunikacji z modulem SCslave to wyłącz dany moduł i zanotuj błąd.
 - 2.3. Sprawdzenie sygnałów od użytkownika lub innych procesów
 - a) Jeżeli zażądano transferu to uruchom równolegle do pracy sterownika program SCwl2dir.
 - b) Jeżeli zażądano zakończenia pracy systemu to wyjdź z obecnej pętli.
3. Jeżeli wystąpił błąd krytyczny wysyłaj sygnał awaryjny, aż nie otrzymasz żądania wyłączenia programu.
4. Uruchom procedurę zakończenia pracy programu.

Komunikacja ze sterownikiem.

Komunikować się ze sterownikiem można za przy pomocy sygnałów systemowych. Istnieją dwa komunikaty, które można wysłać do sterownika:

- Transfer – rozpoczyna transfer danych pobranych z modułów SCslave do katalogu transferowego; komunikat wywołuje się przy pomocy sygnału systemowego SIGUSR1,
- System_off – powoduje zakończenie pracy programu; komunikat wywołuje się przy pomocy sygnałów systemowych SIGTERM lub SIGQUIT.

Konfiguracja programu.

Pierwszym etapem konfiguracji programu SCrunner jest określenie wartości zmiennych zawartych w pliku konfiguracyjnym ./scranner.ini.

Następujące wartości muszą być określone:

- [main]
 - scs_offtest – 1 jeżeli układy SCslave mają być wyłączane po nieudanych przejściu testów początkowych lub po kilku nieudanych próbach odebrania z nich danych, 0 w. p. w.,
 - proclog – 1 jeżeli mają być tworzone pliki typu Proclog, 0 w. p. w.,
 - trans_dir – ścieżka katalogu transferowego,
- [workstations] – w tej sekcji należy zdefiniować zmienne o nazwach w postaci identyfikatorów maszyn i wartościach ścieżek portów COM, pod które podpięte są układy SCslave. Na przykład jeżeli moduł SCslave maszyny o identyfikatorze 4 jest podłączony do portu /dev/ttyS3 to definiujemy zmienną 4=/dev/ttyS3.

Opcjonalne wartości to:

- [delays]
 - tact_microsec – opóźnienie taktów w mikrosekundach,
 - cycle_sec – opóźnienie cyklu pracy w sekundach,
 - cycle_microsec - opóźnienie cyklu pracy w mikrosekundach (obydwie wartości cycle_* są sumowane),

Drugim etapem jest wyedytowanie skryptu Bash exit.sh jeżeli tuż przed zakończeniem pracy programu powinny zostać uruchomione jakieś dodatkowe polecenia. Skrypt ten musi zwracać wartość 0 jeżeli jego wykonanie przebiegło bezproblemowo, w przeciwnym wypadku system zanotuje lekki błąd. Jeżeli przed i/lub po transferze należy wykonać dodatkowe operacje to wtedy należy przygotować skrypty programu SCwl2dir, które są opisane w dokumentacji tego programu (patrz. roz. [4.3.2.SCwl2dir](#)).

Pliki i katalogi.

- ./scranner – plik wykonalny sterownika SRunner,
- ./scwl2dir – plik wykonalny podprogramu SCwl2dir (patrz. roz. [4.3.2.SCwl2dir](#)),
- ./pre-trans.sh - skrypt Bash uruchamiany przez podprogram SCwl2dir przed rozpoczęciem transferu,
- ./post-trans.sh - skrypt Bash uruchamiany przez podprogram SCwl2dir po zakończeniu transferu.
- ./scranner.ini – plik konfiguracyjny,
- ./error.log – błędy zanotowane od ostatniego transferu (patrz roz. [4.7.4.Errorlog \(error.log\)](#)),
- ./exit.sh – skrypt Bash uruchamiany tuż przed zakończeniem pracy programu SRunner,

- ./worklogs/ - katalog przechowujący pliki typu worklog (patrz roz. [4.7.2.Worklog \(*.wl\).](#)),
- ./proclogs/ - katalog przechowujący pliki typu proclog (patrz roz. [4.7.3.Processlog \(*.pl\).](#)).

Opis błędów.

Błędy powstałe podczas pracy sterownika są logowane w plikach typu proclog i pliku ./error.log zgodnie z ich formatami (patrz roz. [4.7.Formaty plików](#)). Kod błędu składa się z dwóch liczb: kod typu błędu oraz kod szczegółowy. Opisy błędów znajdują się w tabeli 4.5.

Typ			
Kod	Nazwa - opis	Kod	Opis
0	Startup <i>Błędy podczas rozruchu programu.</i>	2	Nie można otworzyć pliku ./settings.txt.
		4	Nie można otworzyć pliku ./wl.set.
		6	Zły format pliku ./settings.ini.
		8	Zły format lub błędne dane w wartości w sekcji workstations w pliku ./settings.ini.
		10	Nie można utworzyć lub otworzyć pliku typu worklog.
		12	Nie można przejąć portu COM.
		14	Nie można przechwycić sygnałów systemowych.
1	SCs_nocheck	X	Układ SCslave o numerze X nie działa poprawnie lub nie jest podłączony.
4	Transfer <i>Błędy podczas transferu.</i>	1	Podprogram SCwl2dir wywołany z niepoprawną liczbą argumentów.
		3	Nie można utworzyć pliku wl.set w katalogu transferowym.
		5	Nie można utworzyć kopii pliku ./wl.set o nazwie ./oldwl.set.
		7	Nie można otworzyć pliku ./wl.set
		11	Nie można utworzyć pliku o nazwie terror w katalogu transferowym.
		13	Błąd skryptu pre-transferowego.
		15	Błąd skryptu post-transferowego.
		17	W katalogu transferowym znajdują się pliki z poprzedniego transferu.
		21	Nie można utworzyć nowego procesu dla programu SCwl2dir.
5	Unlogged	X	Nie udało się X razy zalogować danych pobranych z układów SCsalave.

Tab. 4.5. Kody błędów programu SCrunner.

Są dwie rodzaje błędów:

- krytyczne – powodują zakończenie pracy programu; są to błędy typu Startup i Unlogged,
- lekkie – nie powodują zakończenia pracy programu.

4.3.2. SCwl2dir.

Podprogramu sterownika SCrunner odpowiedzialny za transfer danych do katalogu transferowego. Opis zawartość katalogu transferowego znajduje się w rozdziale [4.7.5.Katalog transferowy](#).

Argumenty i zwracane wartości.

Program przyjmuje jeden argument w postaci ścieżki do katalogu transferowego. Program zwraca wartości 0 jeżeli transfer zakończył się sukcesem lub kod szczegółowy błędu typu Transfer jeżeli wystąpił błąd podczas transferu (patrz. roz. [4.3.1.SCrunner](#)).

Procedura transferu.

1. Uruchom skrypt pre-transferowy ./pre-trans.sh.
2. Sprawdź czy katalog transferowy nie zawiera plików po poprzednim transferze. Jeśli są przerwij transfer.
3. Utwórz pusty plik o nazwie terror w katalogu transferowym.
4. Stwórz katalog worklogs w katalogu transferowym.
5. Zmień nazwę pliku wl.set w katalogu programu na oldwl.set.
6. Dla wszystkich dat zawartych w pliku oldwl.set powtarzaj:
 - 6.1. Skopiuj plik worklog odpowiadający danej dacie z katalog_programu/worklogs/ do katalog_transferowy/worklogs/. Jeżeli kopiowanie nie powiedzie się to zapisz daną datę oraz pozostałe daty z pliku oldwl.set do pliku wl.set w katalogu programu i wyjdź z obecnej pętli.
 - 6.2. Utwórz plik sumy kontrolnej MD5 dla pliku worklog odpowiadający danej dacie z katalog_programu/worklogs/ w katalog_transferowy/worklogs/.
 - 6.3. Zapisz daną datę do pliku wl.set w katalogu transferowym.
7. Przenieś plik error.log z katalogu programu do katalogu transferowego.
8. Jeżeli nie wystąpił błąd podczas transferu to usuń plik terror z katalogu transferowego.
9. Uruchom skrypt post-transferowy ./post-trans.sh.

Konfiguracja programu.

Program SCwl2dir wymaga konfiguracji tylko jeżeli przed i/lub po transferze należy dokonać jakiejś dodatkowej operacji. Takimi dodatkowymi operacjami mogą być na

przykład zamontowanie pendrive'a, na którym znajduje się katalog transferowy i jego odmontowanie po zakończeniu transferu. Dostępne są to tego celu dwa skrypty Bash:

- `./pre-trans.sh` - uruchamiany przed rozpoczęciem transferu,
- `./post-trans.sh` - uruchamiany po zakończeniu transferu.

Skrypty muszą zwracać wartość 0 jeżeli ich wykonanie przebiegło bezproblemowo, w przeciwnym wypadku system zanotuje lekki błąd.

Pliki i katalogi.

- `./pre-trans.sh` - skrypt Bash uruchamiany przez program SCwl2dir przed rozpoczęciem transferu,
- `./post-trans.sh` - skrypt Bash uruchamiany przez podprogram SCwl2dir po zakończeniu transferu.

4.3.3. SCrunner_tui.

Program SCrunner_tui jest opartym o ncurses tekstowym front-end'em dla sterownika SCrunner. Dzięki SCrunner_tui można w wygodny sposób śledzić prace sterownika oraz wysłać do niego komunikat żądający transferu danych lub wyłączenia systemu. Jeżeli zostanie w niego wkompilewana biblioteka libscanalis z bibliotek systemu SC to program SCrunner_tui będzie w stanie także przedstawiać w locie wyniki analizy danych pobranych z modułów SCslave. Jest to szczególnie przydatna opcja jeżeli prototypujemy nowy mechanizm analizy danych, który będzie zastosowany w grupie zarządzającej systemu SC.

Obsługa programu.

Program SCrunner_tui przyjmuje jeden argument w postaci ścieżki do pliku lub pipeline'a z wyjściem programu SCrunner. Przykładowe uruchomienie może wyglądać następująco:

```
> mkfifo scr_pipe
> ./scrrunner > scr_pipe
> ./scrrunner_tui scr_pipe
```

Poniższy rysunek 4.6 przedstawia zrzut ekranu programu SCrunner_tui.

Oprogramowanie tej grupy chodzi zarówno na systemach z rodziny Linux, jak i MS Windows. Przenośność została uzyskana poprzez kross-kompilację.

4.4.1. SCwl2db.

SCwl2db jest programem odpowiedzialnym za odbiór danych z katalogu transferowego i przekazanie ich do bazy danych SCdb.

Procedura odbioru transferu.

- 1 Sprawdź czy w katalogu transferowym znajduje się plik o nazwie terror. Jeżeli tak to przejdź do drukowania błędów z pliku error.log.
- 2 Dla każdego pliku typu worklog z dnia określonego w pliku wl.set w katalogu transferowym wykonaj:
 - 2.1 Wydrukuj datę, z której pochodzi plik worklog.
 - 2.2 Sprawdź sumę kontrolną pliku worklog.
 - 2.3 Zanotuj datę pliku worklog w bazie danych.
 - 2.4 Przekaz wszystkie dane z pliku worklog do bazy danych.
- 3 Wydrukuj błędy z pliku error.log.

Konfiguracja programu.

Program posiada w swoim katalogu roboczym plik konfiguracyjny o nazwie scwl2db.ini, w którym muszą być zdefiniowane następujące zmienne:

- [main]
 - trans_dir – ścieżka do katalogu transferowego.

Program SCwl2db korzysta także z pliku konfiguracyjnego scdb.ini (patrz. roz. [4.5.3.scdb.ini.](#)), który także musi być w jego katalogu roboczym.

Obsługa programu.

Transfer danych z katalogu transferowego do bazy danych następuje po wywołaniu programu SCwl2db w konsoli z następującymi parametrami:

scwl2db erase over [trans_dir]

- erase – wartość 1 jeżeli program ma usunąć pliki po transferze, w. p. w. wartość 0,
- over – wartość 1 jeżeli powtarzające się klucze główne mają być nadpisane podczas transferu, w. p. w. wartość 0.
- trans_dir – ścieżka do katalogu transferowego; jeżeli ten parametr nie jest podany to ścieżka katalogu transferowego jest czytana z pliku scwl2db.ini.

4.4.2. SCmanager.

SCmanager jest programem odpowiedzialnym za zarządzanie danymi zwartymi w bazie danych SCdb.

Konfiguracja programu.

Program korzysta z pliku konfiguracyjnego scdb.ini (patrz. roz. [4.5.3.scdb.ini.](#)), który musi znajdować się w jego katalogu roboczym.

Obsługa programu.

Programem steruje się z konsoli i ogólny schemat wywoływania programu wygląda następująco:

scmanager żądana operacja [dodatkowe argumenty ...]

Możliwe operacje oraz ich argumenty są opisane poniżej. Wszystkie operacje z wyjątkiem insert_shift i insert_worker nie pobierają żadnych danych z standardowego wejścia.

- Wykaz dni danego miesiąca, z których baza danych zawiera dane pobrane z układów SCslave:

scmanager get_wls year month

- year – rok,
- month – miesiąc.

- Wykaz pracowników znajdujących się w bazie danych:

scmanager get_workers

- Wstawienie pracownika do bazy danych:

scmanager insert_worker

Następnie na wejście programu należy podać imię i nazwisko pracownika zakończone znakiem nowej linii.

- Usunięcie pracownika z bazy danych.

scmanager insert_worker

Następnie na wejście programu należy podać imię i nazwisko pracownika zakończone znakiem nowej linii.

- Wyświetl plan dnia:
scmanager get_shift date
– date – data w formacie YYYY-MM-DD.

- Ustal plan dni:
scmanager insert_shift date
– date - data w formacie YYYY-MM-DD.

Jest to jedyna operacja czytająca dane z standardowego wejścia. Należy podać ciąg wpisów w formacie:

first_name last_name station shift\n

gdzie:

- first_name – imię pracownika,
- last_name – nazwisko pracownika,
- station – numer stacji robocza, na której pracował,
- shift – numer zmiany, podczas której dany pracownik pracował na danej maszynie.

Po ostatnim wpisie należy zamknąć standardowe wejście.

4.4.3. SCraport.

SCraport jest programem generującym raporty na podstawie danych zawartych w bazie danych SCdb.

Konfiguracja programu.

Program posiada w swoim katalogu roboczym plik konfiguracyjny o nazwie scraport.ini, w którym muszą być zdefiniowane następujące zmienne:

- [shift] – wszystkie zmienne w tej grupie przyjmują jako wartość czas w formacie HH:MM:SS i odnoszą się do czasu trwania poszczególnych zmian z zakładowej pracy,
 - first_start – początek pierwszej zmiany,
 - first_end – koniec pierwszej zmiany,
 - second_start – początek drugiej zmiany,
 - second_end – koniec drugiej zmiany,
 - third_start – początek trzeciej zmiany,
 - third_end – koniec trzeciej zmiany.
- [values] – w tym dziale należy zdefiniować jak mają być podpisane kolumny tabel generowanych w raportach. Np. "2 = 5 cm", czyli kolumna nr 2 zostanie podpisana jako 5 cm.

Program SCraport korzysta także z pliku konfiguracyjnego scdb.ini (patrz. roz. [4.5.3.scdb.ini](#)), który także musi znajdować się w jego katalogu roboczym.

Obsługa programu.

Programem steruje się z konsoli i ogólny schemat wywoływania programu wygląda następująco:

scraport żądana operacja [dodatkowe argumenty ...]

- Generacja raportu typu "Pracownicy – pojedynczy dzień":

scmanager gen_day_report data format

- date - data w formacie YYYY-MM-DD,
- format – format raportu (txt, czyli tekstowy lub html).

Raport jest drukowany do standardowego wyjścia.

- Generacja raportu typu "Pracownicy – pojedynczy miesiąc":

scmanager gen_month_report year month format

- year – rok,
- month – miesiąc,
- format – format raportu (txt, czyli tekstowy lub html).

Raport jest drukowany do standardowego wyjścia.

- Generacja raportu typu "Stanowiska – krótki okres czasu":

scmanager gen_short_report data from_time to_time station format

- date - data w formacie YYYY-MM-DD,
- from_time – czas rozpoczęcia,
- to_time – czas zakończenia,
- station – numer stanowiska,
- format – format raportu (txt, czyli tekstowy lub html).

Raport jest drukowany do standardowego wyjścia.

Typy raportów.

- Pracownicy – pojedynczy dzień

Generuje tabelę 4.7, w której zawarte są wyniki dla wszystkich pracowników z danego dnia.

	Wynik 1	Wynik 2	...
Pracownik A			
Pracownik B			
...			

Tab. 4.7. Tabela raportu.

- Pracownicy – pojedynczy miesiąc

Generuje tabelę 4.7, w której zawarte są wyniki dla wszystkich pracowników z danego miesiąca.

- Stanowisko – krótki okres czasu.

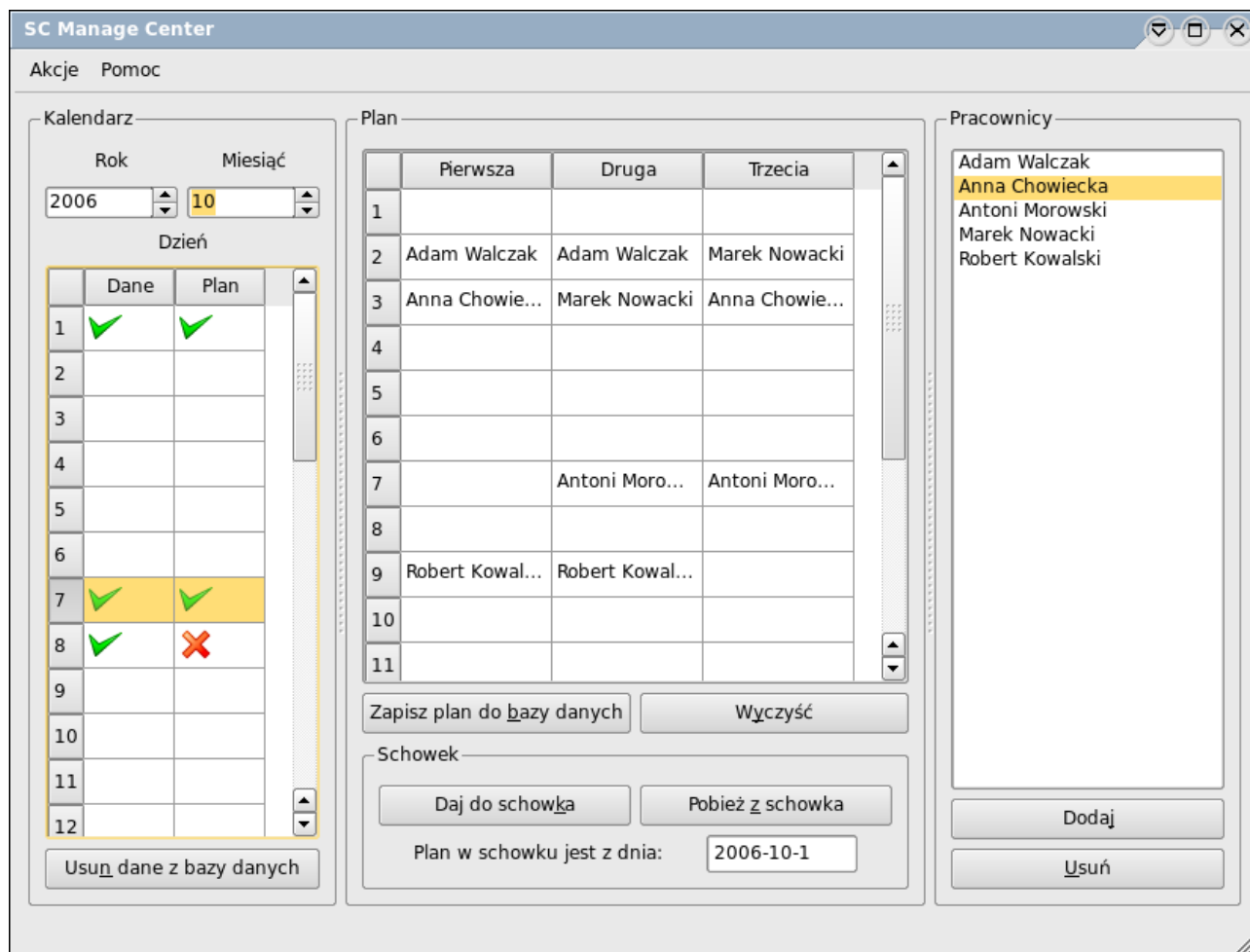
Generuje tabelę 4.8, w której zawarte są wyniki dla danego stanowiska z danego dnia i przedziału czasu.

	Wynik 1	Wynik 2	...
Stanowisko			

Tab. 4.8. Tabela raportu.

4.4.4. SC Manage Center.

Aplikacja SC Manage Center jest front-end'em dla programów klienckich bazy danych SCdb. Zaleca się stosowanie tego programu do obsługi całej grupy zarządzającej systemu SC ponieważ programy konsolowe tej grupy są stosunkowo nie wygodne. SC Manage Center jest całkowicie oparty na platformie Qt 4 firmy Trolltech, co zapewnia dużą przenośność pomiędzy systemami operacyjnymi [3]. Szczegółowy opis obsługi programu znajduje się w "SC Manage Center Manual".



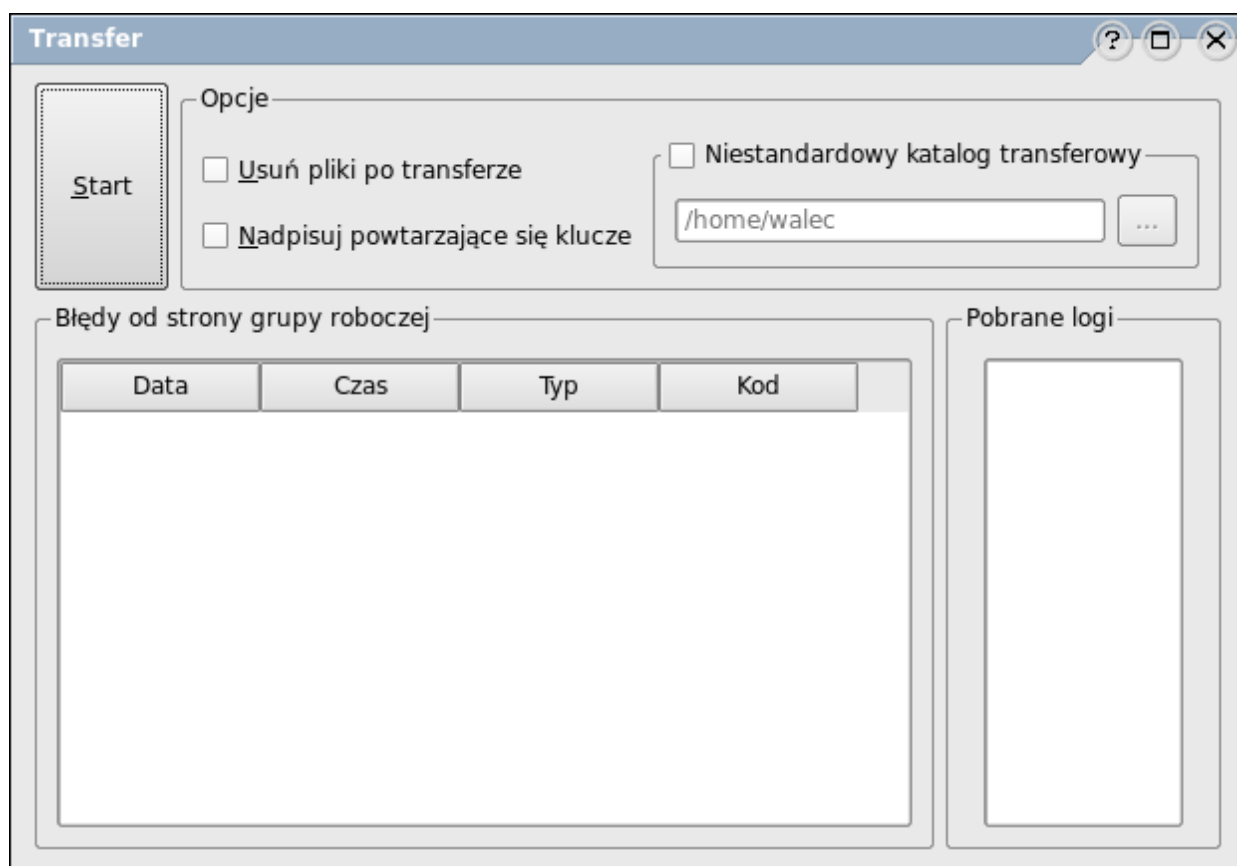
Rys. 4.9. Zrzut ekranu SC Manage Center

Główne okno programu SC Manager Center podzielone jest na trzy rubryki:

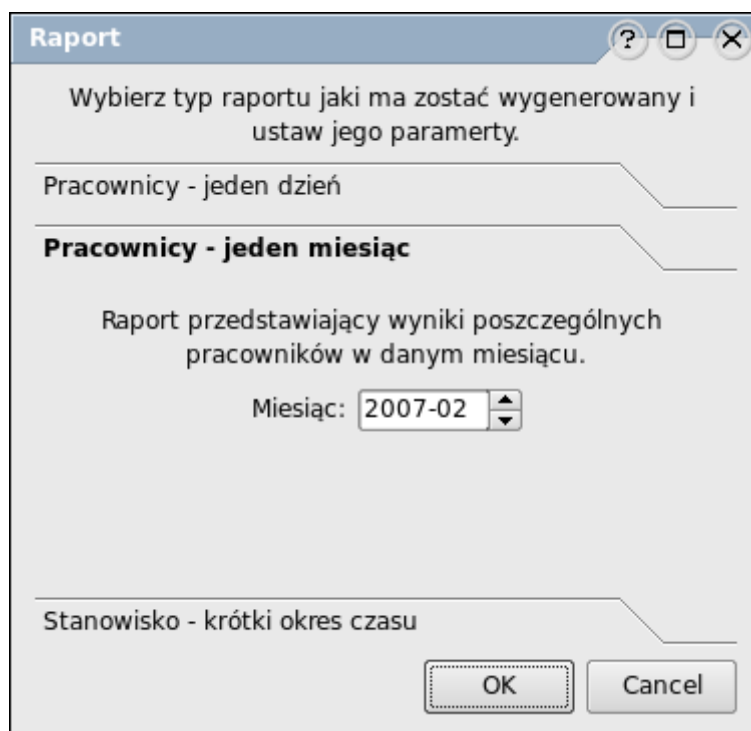
- Kalendarz – przedstawia, w których dniach system rejestrował dane oraz dla których został wprowadzony harmonogram pracy,
- Plan – pole do edycji harmonogramu pracy,
- Pracownicy – spis pracowników.

W głównym menu programu dostępne są dwie akcje:

- Transfer – odbiór transferu danych od grupy roboczej (rys. 4.10),
- Raport – generacja raportów (rys. 4.11).



Rys. 4.10. Zrzut ekranu okna transferu.



Rys. 4.11. Zrzut ekranu okna generowania raportu.

Sterowanie programami klienckimi.

Poszczególne programy klienckie są przypisane do elementów aplikacji SC Manage Center w następujący sposób:

- SCmanager – operuje na elementach głównego okna aplikacji,
- SCtransfer – uruchamiany po wciśnięciu przycisku OK w oknie transferu,
- SCraport – uruchamiany po wciśnięciu przycisku OK w oknie raportu.

Ponieważ program SCmanager jest odpowiedzialny za szeroki wachlarz czynności aplikacja SC Manage Center może wykorzystywać jednocześnie wiele jego instancji. Poszczególne operacje wykonywane przez program SC Manage Center mogą blokować rubryki głównego okna aplikacji aby zapobiec utracie lub przekłamanie wprowadzanych danych. Zależność ta jest przedstawiona w tabeli Tab. 4.12.

Operacja	Blokada rubryki		
	Kalendarz	Plan	Pracownicy
Zmiana daty	X	X	
Usuwanie danych z danego dnia	X	X	
Zapis planu	X	X	X
Dodanie pracownika			X
Usuwanie pracownika	X	X	X

Tab. 4.12. Blokowanie rubryk.

4.5. Baza danych SCdb

Baza danych oparta o MySQL 5.* i jest elementem grupy zarządzającej systemu SC. Źródła bazy danych składają się z dwóch plików, które znajdują się w katalogu ./software/manage_grp/SCdb/:

- Scdb_base.sql – struktury tabel,
- SCdb_proc.sql – przechowywane procedury.

Kodowanie znaków w bazie danych SCdb jest w standardzie UTF-8. W katalogu ./software/manage_grp/SCdb/ znajduje się także przykładowy plik konfiguracyjny scdb.ini wykorzystywany przez programy klienckie bazy danych (patrz roz. [4.5.3.scdb.ini](#)).

4.5.1. Schemat bazy danych.

Schemat logiczny.

worker(id_worker, first_name, last_name)

worklog(id_wl, wl_date, trans_date)

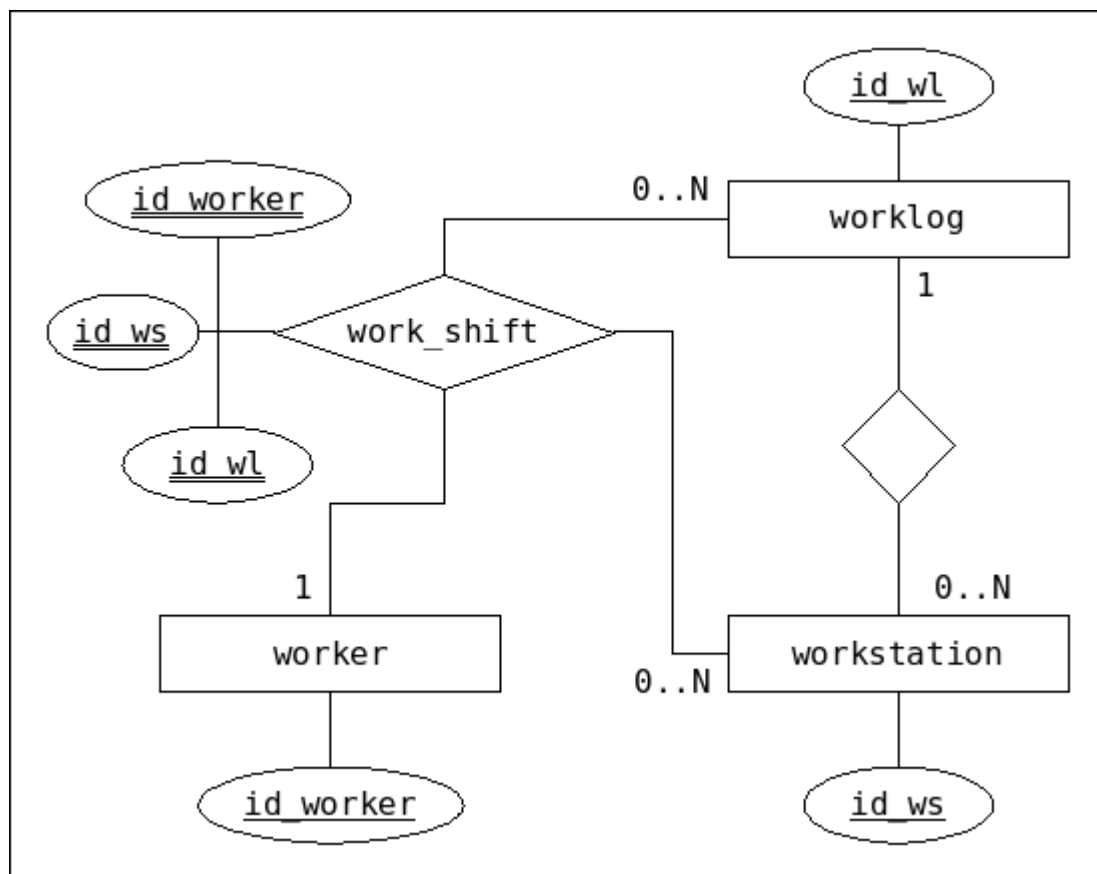
workstation(id_ws, id_sdate, stime, input)

work_shift(id_worker, id_ws, wdate, shift)

Szczegółowy opis poszczególnych relacji i atrybutów:

- worker – dane pracowników
 - id_worker – identyfikator pracownika,
 - first_name – imię,
 - last_name – nazwisko,
- worklog – dane na temat pobranych plików typu worklog z katalogu transferowego (patrz. roz. [4.7.2.Worklog \(*.wl\)](#) i [4.7.5.Katalog transferowy](#).)
 - id_wl – identyfikator daty pliku worklog,
 - wl_date – data pliku worklog,
 - trans_date – data dostarczenia pliku worklog do bazy danych,
- workstation – reprezentuje rekordy danych pobranych z modułów SCslave
 - id_ws – identyfikator modułu SCslave,
 - id_sdate – identyfikator daty pliku worklog, z którego dany rekord pochodzi,
 - stime – czas pobrania danych w modułu SCslave,
 - input – dane z modułu SCslave.
- work_shift – dane dotyczące planu pracy poszczególnych pracowników na poszczególnych stanowiskach
 - id_worker – identyfikator pracownika,
 - id_ws - identyfikator modułu SCslave na stanowisku pracownika,
 - wdate – dzień, w którym dany pracownik pracował na danym stanowisku,
 - shift – numer zmiany, podczas której dany pracownik pracował na danym stanowisku w dniu wdate.

Diagram ER.



Rys. 4.13. Diagram ER.

4.5.2. Procedury.

W przechowywanych procedurach zawarte są wszystkie niezbędne operacje potrzebne dla aplikacji klienckiej i jest zalecane aby korzystać tylko z nich podczas pracy z bazą danych. Opis dostępnych procedur:

- Pracownicy:
 - `show_workers ()`
Wyświetla wszystkich pracowników.
 - `insert_worker (in n_first_name varchar(30), in n_last_name varchar(30))`
Dodaje pracownika o imieniu `n_first_name` i nazwisku `n_last_name`.
 - `delete_worker (in n_first_name varchar(30), in n_last_name varchar(30))`
Usuwa pracownika o imieniu `n_first_name` i nazwisku `n_last_name`.

- Dane z linii produkcyjnej:
 - insert_wl (in n_wl_date date)
Dodaje datę n_wl_date pliku worklog.
 - show_wl (in Ayear int, in Amonth int)
Wyświetla dni, z których dane zawiera baza danych.
 - insert_SCdata (in n_wl_date date, in n_stime time, in n_id_ws tinyint, in n_input int)
Wstawia dane n_input z rejestru układu SCslave z dnia date pobrane w czasie n_stime z stanowiska n_id_ws.
 - insert_SCdata_override(in n_wl_date date, in n_stime time, in n_id_ws tinyint, in n_input int)
To samo co procedura insert_SCdata z tą różnicą, że nadpisuje powtarzające się dane.
 - delete_alldata(in del_date date)
Usuwa wszystkie dane z konkretnego dnia.
- Plan:
 - show_planned_shift (in Ayear int, in Amonth int)
Wyświetla dni dla których jest ułożony plan.
 - insert_shift (in shift_date date, in worker_fname varchar(40), in worker_lname varchar(40), in station int, in shift_num int)
Wstawia element planu, czyli informację, że dnia shift_date pracownik o imieniu worker_fname i nazwisku worker_lname pracował na stanowisku o numerze station podczas zmiany numer shift_num.
 - show_shift (in sdate date)
Wyświetla plan w formie tabeli o kolumnach: imię, nazwisko, numer stanowiska, numer zmiany.
 - delete_shift (in sdate date)
Usuwa plan z dnia date.

- Łączenie danych z planem:
 - show_work (in n_first_name varchar(40), in n_last_name varchar(40), in idate date)
Wyświetla tabelę w formacie: numer stanowiska, numer zmiany; dla pracownika o imieniu n_first_name i nazwisku n_last_name z dnia idate.
 - show_day_input (in sdate date, in tfrom time, in tto time, in ws int)
Wyświetla tabelę w formacie: dane z rejestru SCslave, czas ich pobrania; z dnia sdate, w przedziale czasu od tfrom do tto z stanowiska o numerze ws.

4.5.3. scdb.ini.

Plik scdb.ini jest plikiem konfiguracyjnym wymaganym przez wszystkie programy klienckie dla bazy danych SCdb. Następujące zmienne muszą być w nim zdefiniowane:

- [sql]
 - host – adres serwera, na którym znajduje się baza danych SCdb,
 - user – nazwa użytkownika,
 - passwd – hasło użytkownika,
 - db_name – nazwa bazy danych.

4.6. Kody źródłowe

Wszystkie pliki źródłowe bibliotek i programów znajdują się w katalogu ./software, a ich ogólne charakterystyki przedstawia poniższa tabela 4.14.

Języki programowania	C, C++, Bash script, SQL
Wykorzystane API/Biblioteki	POSIX, MySQL C API, Qt 4, iniparser
Kodowanie znaku końca linii	format UNIX
Kodowanie znaków	ASCII, oprócz źródeł programów grupy zarządzającej, które są kodowane w UTF-8
Kompilatory	GNU GCC lub MinGW (tylko dla gr. zarz. na Win32)
System kompilacji	CMake
Generator dokumentacji	Doxygen

Tab. 4.14. Charakterystyki kodów źródłowych.

Źródła programów z grup roboczej i grupy zarządzającej znajdują się odpowiednio w katalogach ./software/work_grp i ./software/manage_grp. W katalogu ./software/lib znajdują się dodatkowe biblioteki na stałe dostarczane z systemem SC. Jedną z nich jest „Selection Check Libraries”, czyli zestaw bibliotek napisanych specjalnie dla oprogramowania systemu SC.

Dokumentacja kodów źródłowych jest pisana w języku angielskim. Biblioteka SC zawiera automatycznie generowaną dokumentację w postaci HTML, która znajduje się w „./documents/SC Libraries Reference HTML/index.html”. Dokumentacja Selection Check Librarys znajduje się także w postaci pliku PDF w katalogu ./documents/, jednak zalecam przeglądanie dokumentacji HTML'owej. W katalogach źródeł programów i bibliotek są także różne pliki tekstowe zawierające informacje na temat ich wymagań, instalacji oraz ich charakterystyk.

Wszystkie źródła są kompilowane przy pomocy kompilatorów GNU GCC lub MinGW (tylko dla grupy zarządzającej na MS Windows) za pośrednictwem systemu kompilacji CMake.

4.7. Formaty plików

Podczas czytania tego rozdziału należy pamiętać o następujących konwencjach stosowanych przy opisywaniu struktury plików:

- pliki binarne opisywane są za pomocą struktur z języka C,
- znaki specjalne w plikach tekstowych zaznaczone są zgodnie z konwencjami stosowanymi w procedurach printf, scanf, itp. standardowej biblioteki stdio języka C,
- zmienne w plikach tekstowych są podkreślone; ilość znaków wartości zmiennej jest równa długości nazwy zmiennej lub jest nieokreślona jeżeli nazwa zmiennej zaczyna się od znaku '%’.

4.7.1. Worklog's set (wl.set).

Worklogs set jest plikiem tekstowym w standardzie ANSI. Plik ten zawsze występuje wraz z katalogiem worklogs/, który zawiera pliki typu worklog. Struktura pliku jest ciągiem rekordów typu:

DD-MM-YYYY\n

Legenda:

- DD – dzień,
- MM – miesiąc,
- YYYY – rok,

Ciąg dat określa z jakich dni pochodzą pliki worklog z katalogu ./worklogs/ (patrz. roz. [4.7.2. Worklog \(*.wl\)](#)). Data DD-MM-YYYY jest zgodna z formatem zastosowanym w MySQL i można ją bezpośrednio traktować jako zmienną typu DATE.

4.7.2. Worklog (*.wl).

Worklog jest plikiem binarnym stosowanym do przechowywania danych pobranych z układów SCslave. Zawiera ciąg rekordów typu:

```
struct mach_data
{
    struct tms time;
    unsigned int code;
    unsigned int input;
};
```

Legenda:

- time – czas pobrania danych o strukturze:

```
typedef unsigned char tiny_int;
struct tms
{
    tiny_int hour;
    tiny_int min;
    tiny_int sec;
};
```

- code – identyfikator modułu SCslave,
- input – dane pobrane z modułu SCslave.

Poza strukturą pliku worklog ściśle określona jest także jego nazwa, która jest w formacie:

DD-MM-YYYY.wl

Legenda:

- DD – dzień,
- MM – miesiąc,
- YYYY – rok,

Data DD-MM-YYYY określa dzień, z którego pochodzą dane. Format daty jest zgodny z formatem stosowanym w pliku wl.set.

4.7.3. Processlog (*.pl).

Processlog jest plikiem tekstowym w formacie ASCII zawierający log działania programu SCrunner. Zawiera ciąg rekordów typu:

- hh:mm:ss DATA:\tcode:%code\tinput:%input\n
- hh:mm:ss USER:\t%signal\n
- hh:mm:ss ERROR:\t%type:%errno\n

Legenda:

- hh – godzina,
- mm – minuta,

- ss – sekunda,
- %code – kod układu SCslave,
- %input – binarna reprezentacja danych pobranych z układu SCslave,
- %signal – sygnał podany przez użytkownika; możliwe wartości to:
 - System_on – rozpoczęcie programu,
 - Transfer_start – rozpoczęcie transferu,
 - Transfer_end – zakończenie transferu,
 - System_off – zakończenie programu,
- %type – typ błędu; możliwe wartości opisane są w rozdziale [4.3.1.SCranner](#),
- %errno – kod błędu; możliwe wartości opisane są w rozdziale [4.3.1.SCranner](#).

4.7.4. Errorlog (error.log).

Errorlog jest plikiem binarnym przechowujący informacje na temat błędów, które wystąpiły podczas działania programu SCrunner (patrz. roz. [4.3.1.SCranner](#)). Struktura plików to zbiór rekordów typu:

```
struct SCerrno
{
    time_t time;
    int type;
    int code;
};
```

4.7.5. Katalog transferowy.

Opisane są tutaj pliki i katalogi, które mogą znaleźć się po dokonaniu transferu w katalogu transferowym.

- ./terror – pusty plik, który oznacza, że podczas transferu zaistniał poważny błąd,
- ./wl.set – określa pliku worklog, które znajdują się w katalogu worklogs,
- ./worklogs/ - katalog zawierający pliki worklog i pliki z ich sumami kontrolnymi MD5,
- error.log – plik z błędami powstałymi od ostatniego transferu.

5. Wdrożenie systemu

System Selection Check podzielony jest na dwie grupy, które wdrażane są oddzielnie. Łącznikiem obydwu grup jest katalog transferowy.

5.1. Wdrożenie grupy roboczej

Pierwszym etapem wdrażania grupy roboczej jest przygotowanie urządzeń, z których będą pobierane dane poprzez moduły SCslave. Do wejść I 0-7 każdego modułu należy podpiąć dane binarne generowane przez zaadaptowaną maszynę. Należy także ustalić sygnał CK modułu, którego wyzwolenie spowoduje zarejestrowanie danych w module i zaznaczenie ich jako nową porcję danych. Następnie należy podłączyć moduły SCslave do serwera produkcyjnego. Przykład adaptacji skracarki znajduje się poniżej w roz. [Adaptacja skracarki w firmie Walczak](#).

Drugim etapem jest instalacja i konfiguracja oprogramowania serwera produkcyjnego. Całe to oprogramowanie oraz pliki potrzebne do jego poprawnego działania znajdują się w katalogu `./software/work_grp/`. Aby skompilować to oprogramowanie należy wpisać w konsoli:

```
> cmake  
> make
```

W pierwszej kolejności należy zainstalować w systemie biblioteki Selection Check Librarys. Można to zrobić w następujący:

```
> mv ./software/work_grp/*.so /usr/lib/
```

lub jeżeli chcemy wykorzystać już skompilowane pliki

```
> mv ./software/work_grp/package/*.so /usr/lib/
```

i na końcu

```
> ldconfig
```

Następnie należy skopiować wszystkie pliki wykonalne z katalogu `./software/work_grp` lub `./software/work_grp/package` do katalogu, w którym będzie pracował sterownik SCrunner. Tym katalogiem może być na przykład `/opt/scranner`. W tym katalogu należy także

stworzyć katalogi o nazwach worklogs i proclogs.

Konfiguracja sterownika SCrunner polega na odpowiednim wyedytowaniu parametrów zawartych w pliku scrunner.ini (patrz. roz. [4.3.1.SCranner](#)). W tym pliku konfiguracyjnym znajduje się także parametr określający ścieżkę do katalogu transferowego. Jeżeli przed i po zakończeniu transferu należy wykonać pewne dodatkowe operacje aby przygotować katalog transferowy (na przykład zamontować i odmontować pendrive'a) to należy wyedytować odpowiedni skrypty Bash programu SCwl2dir (patrz. roz. [4.3.2.SCwl2dir](#)).

Adaptacja skracarki w firmie Walczak



Zdj. 5.1. Adaptacja skracarki.

Adaptacja skracarki została wykonana przez Andrzeja Rodziaka. W prawym górnym rogu przedstawiona jest seria czujników wykrywających rozmiar wycinanego fryzu. W prawym dolnym rogu czujnik rejestrujący zdarzenie wycięcia fryzu. Po lewej stronie widnieje okablowanie czujników podłączone do układu SCslave.

5.2. Wdrożenie grupy zarządzającej

Pierwszym etapem wdrażania grupy zarządzającej jest przygotowanie bazy danych MySQL w wersji 5 lub wyższej, z którą będą łączyć się programy klienckie bazy danych SCdb (patrz. roz. [4.5.Baza danych SCdb](#)). Na bazie danych MySQL należy wykonać dwa skrypty z katalogu ./software/manage_grp/SCdb: SCdb_base.sql i SCdb_proc.sql.

Dla systemów z rodziny MS Windows pliki wykonalne oraz biblioteki DLL znajdują się w katalogu ./software/manage_grp/packages/win32/ i po wyedytowaniu plików konfiguracyjnych są gotowe do pracy. W systemach z rodziny Linux przed edycją plików

konfiguracyjnych należy zainstalować biblioteki Selection Check Libraries oraz pliki binarne analogicznie jak to miało miejsce w przypadku grupy roboczej.

Najważniejszy plikiem konfiguracyjnym jest plik scdb.ini, który określa parametry dostępu do bazy danych Scdb (patrz. roz. [4.5.3.scdb.ini.](#)). Ostatnim etapem jest konfiguracja programu SCraport (patrz. roz. [4.4.3.SCraport.](#)) oraz programu SCwl2db (patrz. roz. [4.4.1.SCwl2db.](#)). W pliku konfiguracyjnym programu SCwl2db znajduje się także parametr z ścieżką do katalogu transferowego, z którego będą pobierane dane od grupy roboczej systemu.

Użytkownik końcowy grupy zarządzającej systemem powinien ograniczyć się do pracy wyłącznie z programem SC Manage Center (patrz. roz. [4.4.4.SC Manage Center.](#) i „SC Manage Center Manual”). Zawarte są w nim wszystkie opcje programów klienckich bazy danych SCdb i nie wymaga od użytkownika znajomości pracy w konsoli.

Przykładowy raport generowany w firmie Walczak

W firmie Walczak raporty na ogół określają jakiej długości fryzy dany pracownik wyciął w danym okresie czasu. Zatem plik konfiguracyjny programu SCraport (patrz. roz. [4.4.3.SCraport.](#)) określa długości tych fryzów, a sekcja *values* ma następującą postać:

[values]

0 = X ;wyniki odrzucone

1 = 260 mm

2 = 280 mm

3 = 350 mm

4 = 400 mm

5 = 450 mm

6 = 500 mm

7 = >500 mm

Na rys. 5.2 przedstawiony jest render raportu typu „Pracownicy - pojedynczy miesiąc” z marca 2007 roku generowanego do formatu HTML.

Typ raportu: Pracownicy - pojedynczy miesiac.

Data: 2007-3

	X	260 mm	280 mm	350 mm	400 mm	450 mm	500 mm	> 500 mm
Adam Walczak	26	58	30	31	21	17	9	0
Marek Nowacki	289	539	480	279	311	108	81	13
Anna Chowiecka	358	642	551	253	221	94	43	7
Robert Kowalski	0	0	0	0	0	0	0	0
Antoni Morowski	137	321	187	104	115	59	27	2
Kamil Kózka	492	1268	1004	893	640	539	133	71

Plik wygenerowany przez SCraport

Rys. 5.2. Render raportu w formacie HTML.

6. Podsumowanie

Prace nad systemem Selection Check zostały rozpoczęte na początku roku 2005 i trwały półtora roku. Latem roku 2006 została wdrożona pierwsza działająca wersja systemu w zakładzie firmy Walczak (www.walczakparkiety.pl). Do tej pory została zaadaptowana jedna skracarka do testów systemu, które wypadły pomyślnie i spełniły podstawowe założenia systemu. W marcu 2007 projekt Selection Check zakwalifikował się do finału konkursu „Best Student Project” organizowanego przez firmę Power Media (www.power.com.pl) i Politechnikę Częstochowską.

Architektura systemu pokazuje, że mimo upływu lat wzorce projektowe zaczerpnięte z oprogramowania systemów Unix'owych nadal dobrze sprawdzają się w budowie modularnych systemów informatycznych. Są to takie wzorce jak na przykład podział zadań systemu na zbiór mniejszych programów konsolowych lub wymiennosc interfejsów przy pomocy zastosowania front-end'ów.

Ponadto realizacja systemu Selection Check prezentuje sporą liczbę narzędzi dostępnych na zasadach licencji GNU GPL oraz sposobu ich wykorzystania w budowie systemów informatycznych.

Zawartość płyty CD

- **changelog.txt** – opis zmian
- **README.pdf** – ogólne informacje o projekcie
- **Praca inżynierska.pdf** – niniejsza praca inżynierska
- **./documents/** - główna dokumentacja
 - **editable/** – katalog z plikami edytowanymi, z których powstała część dokumentacji
 - **SC Libraries Reference HTML** – opis bibliotek SC w formacie HTML
 - **index.html** – strona startowa dokumentacji
 - **SC Libraries Reference.pdf** – opis bibliotek SC w formacie PDF
 - **SC Manage Center Manual.pdf** – instrukcja obsługi programu SC Manage Center
 - **SC Work Group Walczak Manual.pdf** – instrukcja obsługi grupy roboczej w zakładzie firmy Walczak
 - **Selection Check Dokumentacja Dewelopera.pdf** – dokumentacja systemu
 - **Selection Check Prezentacja.pdf** – prezentacja systemu
- **./hardware/** - pliki źródłowe sprzętu
 - **scslave/** - pliki modułu SCslave
 - **Makefile** – plik budujący wewnętrzny sterownik
 - **modul.c** – źródło wewnętrznego sterownika
 - **scslave.brd** – schemat płytki
 - **scslave.sch** – schemat logiczny
- **./software/** - pliki źródłowe oprogramowania
 - **author** – informacje o autorze
 - **cleaner.sh** – skrypt ułatwiający konserwację źródeł
 - **CMakeListsInc.txt** – plik budujący systemu CMake, z ogólnymi informacjami na temat budowanego systemu
 - **lib/** - biblioteki dostarczane wraz w oprogramowaniem SC
 - **iniparser/** – biblioteka do parsowania plików INI
 - **wlcstdc/** – własna biblioteka ogólnego użytku
 - **sc/** - integralna biblioteka systemu SC
 - **CMakeLists.txt** – plik budujący systemu CMake
 - **Doxyfile** – plik konfiguracyjny dla generatora dokumentacji Doxygen
 - **doc/** – dokumentacja generowana przez Doxygen
 - **src** /- źródła
 - **test/** – narzędzia do testów

- **work_grp/**: - pliki źródłowe oprogramowania grupy roboczej
 - **CMakeLists.txt** – plik budujący systemu CMake
 - **package/** – dystrybucja oprogramowania gr. roboczej dla sys. Linux
 - **scranner/** – źródła programu SCrunner
 - **scranner_tui/** – źródła programu SCrunner_tui
 - **scstest/** – źródła narzędzia do testów
 - **scwl2dir/** – źródła programu SCwl2dir
- **manage_grp/**: - pliki źródłowe oprogramowania grupy zarządzającej
 - **CMakeLists.txt** – plik budujący systemu CMake
 - **scdb/** – źródła bazy danych SCdb
 - **scmanagecenter/** – źródła programu SC Manage Center
 - **scmanager/** – źródła programu SCmanage
 - **scraport/** – źródła programu SCraport
 - **scwl2db/** – źródła programu SCwl2db
 - **package/**: – dystrybucje oprogramowania
 - **win32/** - dystrybucja dla sys. MS Windows
 - **linux/** - dystrybucja dla sys. Linux

Literatura

- [1] Serial Programming HOWTO (for Linux), Gary Frerking,
<http://tldp.org/HOWTO/Serial-Programming-HOWTO/index.html>
- [2] Advanced Linux Programming, Mark Mitchell, Jeffrey Oldham, Alex Samuel, of
CodeSourcery LLC, ISBN 0-7357-1043-0
- [3] Qt Reference Documentation (Open Source Edition),
<http://doc.trolltech.com/4.2/index.html>
- [4] CMake documentation, <http://www.cmake.org/HTML/Documentation.html>
- [5] MySQL 5.1 Reference Manual, <http://dev.mysql.com/doc/refman/5.1/en/>