**\*\*Zabezpieczenie Antywłamaniowe\*\***

**\*\*Dane osobowe:\*\***

**\* Jakub Serwatka, III rok Mikroelektronika w Technice i Medycynie, grupa 2, kubaserw@gmail.com**

**\* Maciej Zdechlikiewicz,​ III rok Mikroelektronika w Technice i Medycynie, grupa 2, maciek.zdechlikiewicz@gmai.com**

**\*\*Data oddania projektu:\*\***

**20 stycznia 2017**

**\*\*Cel programu:\*\***

**Zaprojektowanie programu z graficzną wizualizacją,​ który ma symulować zachowanie systemu antywłamaniowego.**

**\*\*Opis i schemat struktury logicznej programu:\*\***

**Główną procedurą programu jest Alarm, która zawiera w sobie obiekt chroniony Code\_validator i 3 zadania, wykonujące się współbieżnie - Countdown\_task z dwoma wejściami Start i Stop, Keyboard Control oraz Display\_All. To pierwsze odpowiedzialne jest za odliczanie zadanej liczby sekund (15) i sprawdza, czy nie doszło do przerwania w postaci wpisania kodu w postaci String z klawiatury i naciśnięcia klawisza 'enter'. Display\_All służy do wyświetlania programu i wszystkich zdarzeń. Zawiera w sobie procedury:**

**\* Display - do uproszczenia kodu innych procedur poprzez podanie odpowiednich współrzędnych;**

**\* Display\_Banner - rysuje strukturę graficzną programu, przed przejściem do kolejnych stanów;**

**\* Display\_String - wyświetla w odpowiednim miejscu podany string.**

**Keyboard\_control jest zadaniem, które obsługuję przyjmowanie znaku z klawiatury i posiada w swoim ciele maszynę odpowiednich stanów (Standby, Ready, Countdown i Ringing). Jest również odpowiedzialne za sprawdzenie poprawności kodu i ustawienie odpowiedniego stanu.**

**W skład obiektu chronionego Code\_validator wchodzi procedura chroniona Insert\_Digit i Code\_Reset oraz funkcja chroniona Get\_inserted\_Code. Pierwsza z nich zajmuje się obsługą wpisywania kodu, włącznie z klawiszami backspace i enter. Druga służy do czyszczenia zmiennych po resecie. Get\_inseted\_code zwraca wartość w postaci string. W ciele procedury Alarm pojawia się tylko pusta pętla. W projekcie znajdują się również 3 flagi: Exit\_program, Code\_Inserted i Running zadeklarowane jako zmienne atomowe, aby zapewnić poprawną komunikację pomiędzy zadaniami. Schemat logiczny aplikacji zamieszczony został poniżej:**

**http://home.agh.edu.pl/~jpi/dokuwiki/lib/exe/fetch.php?w=119&h=84&t=1485046618&tok=2700c2&media=dydaktyka:ppscr:2016:schemat\_logiczny.jpg**

**\*\*Informacje o pakietach zewnętrznych:​\*\***

**W projekcie wykorzystano następujące pakiety zewnętrzne:**

**\* Text\_IO,**

**\* Integer\_Text\_IO,**

**\* NT\_Console,**

**\* Strings.Fixed,**

**\* Real\_Time;**

**\*\*Instrukcja obsługi:\*\***

**\* Program rozpoczyna się w trybie '​Standby'​. Jest on odpowiednikiem wyłączonego alarmu i jest reprezentowany za pomocą świecącej się czerwonej diody.**

**\* Po naciśnięciu przycisku '​s'​ aktywujemy alarm. W tym stanie jest on uzbrojony i załącza się zielona dioda.**

**\* Po naciśnięciu przycisku '​a'​ następuje symulacja włamania. Wtedy włącza się licznik odmierzający 15 sekund. W tym czasie można wpisać kod rozbrajający alarm (7536). Kod zostaje zatwierdzony po wciśnięciu przycisku 'enter'.**

**\* Jeśli czas na wpisanie alarmu upłynie, bądź wpisze się zły kod, to uruchamia się alarm, który jest przedstawiony w postaci czerwonego dzwonka.**

**\* Alarm można zresetować po naciśnięciu przycisku '​r'​.**

**\*\*Możliwe rozszerzenia programu: ​\*\***

**\* Wysyłania informacji o włamaniu do policji.**

**\* Zapisywanie do pamięci informacji o dokładnej dacie i godzinie aktywowania alarmu.**

**\* Efekty dźwiękowe podczas wywołania alarmu.**