Architektura Komputerów 2
Projekt

Hardware Keylogger

Oskar Gusta 263970

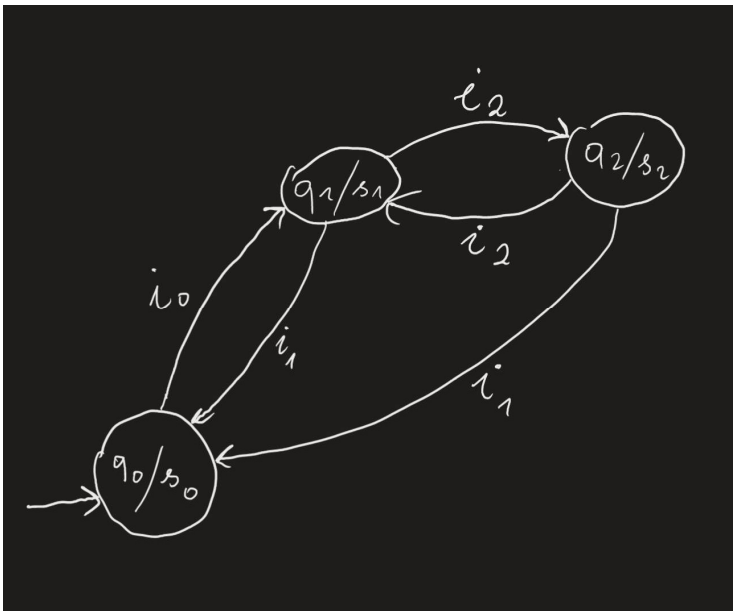
Jakub Kolbertowicz 263988

Łukasz Mróz 263903

1. Cel projektu

Projekt miał na celu poznanie sposobu komunikacji z urządzeniami peryferiów. Należało przygotować urządzenie oparte o wybrany mikrokontroler z samodzielnie przygotowanym oprogramowaniem służącym do podsłuchiwania i zapisywania naciskanych na klawiaturze klawiszy. Urządzenie powinno być podłączane pomiędzy fizyczną klawiaturą a komputerem.

2. Sposób wykonania zadania



Projekt przygotowany został najpierw jako graf, następnie na jego podstawie przystąpiono do przygotowania rozwiązania. Stany q0, q1 i q2 to odpowiednio brak zasilania i połączenia, urządzenie z zasilaniem i połączeniem oraz urządzenie w trybie odczytu zapisanych danych. Wejścia oznaczają: i_0 - podłączenie klawiatury do urządzenia oraz urządzenia do komputera, i_1 - utrata połączenia, i_2 - wciśnięcie kombinacji klawiszy służącej do sprawdzania zapisanych danych. Na grafie zaznaczone są również wyjścia: s_0 to brak wyjść, s_1 to zapis znaków w pamięci urządzenia, s_2 to odczyt zapisanych znaków.

Projekt oparty jest na dwóch mikrokontrolerach: Arduino Pro Micro oparty o chip ATmega32U4 oraz Raspberry Pi Zero. Kod pisany jest w języku C++ dla Arduino oraz Python dla Raspberry Pi. Na początku rozważaliśmy przygotowanie urządzenia z użyciem tylko jednego mikrokontrolera. Powstaje tu jednak problem - do keyloggera potrzebny jest host i urządzenie USB - host USB musi zarządzać komunikacją z klawiaturą i przyjmować od niej sygnały, urządzenie USB musi wysyłać sygnały do komputera. ESP32 czy Arduino, których użycie rozważaliśmy, nie są w stanie spełniać obu tych funkcji. Istnieje możliwość użycia urządzenia USB Host Shield, jednak z powodów finansowych (urządzenie to jest drogie i żaden z nas go nie posiada) zdecydowaliśmy się wykorzystać inną możliwość - skorzystać z posiadanego już przez nas Raspberry Pi, który przejąłby funkcję hosta USB, a także zajmowałby się zapisywaniem klawiszy.

Część wykorzystująca Raspberry Pi pobiera znaki przy pomocy biblioteki getch. Następnie znaki są przekazywane do portu szeregowego mikrokontrolera z wykorzystaniem biblioteki Serial oraz zapisywane w pamięci urządzenia. Takie kody odbiera drugi z mikrokontrolerów. Wykorzystywana jest biblioteka Keyboard.h pozwalająca emulować wciśnięcia znaków. Arduino przyjmuje znaki z portu szeregowego i wysyła odpowiednie wciśnięcia do portu USB komputera.

Aby zakończyć działanie programu i uzyskać dostęp do pliku ze znakami należy wcisnąć odpowiednią kombinację klawiszy.

Raspberry Pi i Arduino połączone zostały ze sobą w następujący sposób:

RPi	**Arduino**
5v	VCC
GPIO14/TXD	RX
GPIO15/RXD	TX
GND	GND

Do działania oprócz napisanego kodu i podłączenia przewodów potrzebne było odpowiednie przygotowanie Raspberry Pi oraz Arduino. Na Raspberry najpierw zainstalowany został system Raspberry Pi OS (znany również jako Raspbian). Udało się to zrobić dzięki zastosowaniu Raspberry Pi Imager. Następnie należało włączyć port szeregowy i autologowanie. Normalnie port szeregowy w Linuxie jest używany przez shell, więc poprzez raspi-config i wyłączenie opcji, która to powoduje, port szeregowy został dopuszczony do użytku. Autologowanie również włączono edytując potrzebne opcje w raspi-config. Kolejnym krokiem było wpisanie ścieżki napisanego programu do /etc/profile, co powoduje automatyczne uruchomienie programu po zalogowaniu. Przechwycone klawisze zapisują się we wcześniej zdefiniowanym pliku w systemie, dzięki czemu można się do nich dostać poprzez ssh, wkładając kartę pamięci do komputera lub podłączając monitor do Raspberry.

Aby przygotować Arduino, należy zainstalować na komputerze platform.io oraz CLion. Następnie należy wejść w terminalu do folderu projektu i zinicjalizować projekt wpisując komendę *pio project init -b leonardo --sample-code -ide clion.* Budowanie projektu odbywa się za pomocą komendy *pio release*, a flashowanie na Arduino komendą *pio upload*.

Potencjał na rozwój widoczny jest w liczbie obsługiwanych kombinacji klawiszy. Jako że każda z nich musi być zaimplementowana pojedynczo, urządzenie nie obsługuje każdej możliwej kombinacji klawiszy typu CTRL+klawisz. Zdecydowanie można też zmniejszyć urządzenie - istnieją takie keyloggery, które są wielkości pendrive'a, jednak są one produkowane masowo z odpowiednich komponentów, nie z tego co jest akurat pod ręką jak w przypadku naszego projektu. Istnieje również wcześniej wspomniana możliwość użycia USB Host Shield. Zwiększyłoby to koszt urządzenia, ale zmniejszyłoby jego rozmiar i usunęłoby jedną z większych wad - stosunkowo długi czas uruchomienia.

3. Analiza zagrożenia, metody przeciwdziałania

Największym zagrożeniem związanym z tego typu urządzeniem jest możliwość przechwycenia wszystkich naciśnięć przycisków na klawiaturze. Mając takie informacje można odtworzyć wpisywane hasła, dane karty płatniczej czy wrażliwe dane osobowe. O ile w przypadku przechwycenia hasła czy danych karty można obronić się coraz powszechniejszą weryfikacją dwuetapową np. przy pomocy aplikacji na telefon, to w przypadku wrażliwych danych osobowych (adres, nr PESEL) nie jest to już tak łatwe. Najłatwiejszą metodą obrony przed hardware keyloggerem jest upewnianie się, że między klawiaturą a komputerem nie ma żadnego urządzenia pośredniczącego, lub że urządzenie typu przedłużacz lub hub jest zaufane i bezpieczne.