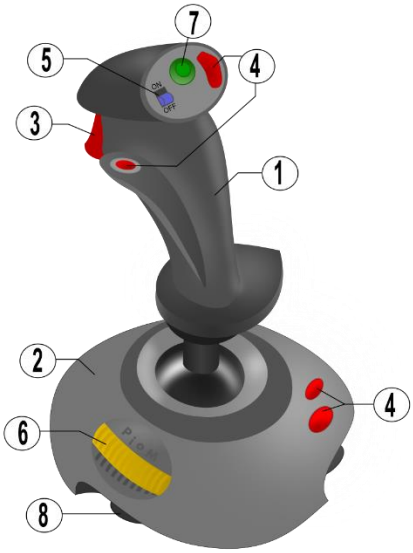
## Zasada działania joysticka

Joystick przekazuje dane na temat ruchu i akcji użytkownika do komputera, z którym jest połączony. Fundamenty działania joysticka to:

- **Osie ruchu** – joystick zawiera osie X oraz Y, które odpowiadają za ruch w płaszczyźnie pionowej. Może istnieć także dodatkowa oś Z, najczęściej odpowiadająca za obroty. Przesunięcie drążka wzdłuż tych osi generuje sygnały analogowe, które następnie przekazywane są do komputera.
- **Czujniki** – joystick posiada zestaw czujników, które monitorują ruch i pozycję drążka oraz stany przycisków.
- **Pozycjometry** – czujniki pozwalające na dokładnie określenie kierunku, w którym skierowany jest drążek.
- **Przyciski** -joysticki generują sygnały cyfrowe w momencie wciśnięcia bądź zwolnienia. Stan przycisków może być odczytany przez komputer i użyty do wyzwalania różnorodnych akcji.
- **Komunikacja z komputerem** – joystick komunikuje się z komputerem najczęściej za pomocą kabla USB. Przekazywane przez niego są dane zebrane przez czujniki.
- **Sterowniki oraz API** – właściwa interpretacja danych z joysticka wymaga obecności odpowiednich sterowników. Oprogramowywanie działania joysticka wymaga również specjalnych interfejsów programistycznych, takich jak DirectInput dla systemów Windows.




## DirectX

Zbiór interfejsów programistycznych (API), stworzonych przez firmę Microsoft, pozwalających na obsługę różnego rodzaju multimediów na platformach tejże firmy (Windows, Xbox).

W celu rozpoczęcie pracy z DirectX należy wyposażyć się w DirectX Software Development Kit (SDK), zawierający biblioteki DirectX w formie binarnej wraz z dokumentacją i plikami nagłówkowymi. Początkowo SDK musiało być instalowane manualnie przez użytkowników bądź w ramach instalacji używającego DirectX programowania. Od momentu wydania Windows 8 SDK jest integralną częścią Windows Development Kit. Wraz z rozwojem frameworku .NET stworzył specjalne, opakowane i przyjazne wersje DirectX nazywane MDX (Managed DirectX) i XNA, które mogą być używane w językach takich jak C++ czy C#.

### DirectInput

Biblioteka będąca częścią składową DirectX (obecnie porzuconą). Pozwala ona użycie w programowanych na system Windows aplikacjach danych z urządzeń wejściowych takich jak: myszka, klawiatura, gamepad bądź joystick. Dodatkowo pozwala na mapowanie przycisków i innych elementów urządzeń wejścia na zdefiniowane przez użytkownika akcje (*Action mapping*).

Link do dokumentacji:
https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ee416842(v=vs.85)?redirectedfrom=MSDN

### Hardware Abstraction Layer (HAL)

Warstwa abstrakcji sprzętowej w systemach komputerowych. Najczęściej jest to oprogramowanie bądź interfejs, umożliwia programistom dostęp do sprzętu bez konieczności zwracania uwagi na specyfikę danego modelu urządzenia. Osiąga się to poprzez ukrycie zbędnych dla programistów szczegółów implementacyjnych oraz różnic sprzętowych między różnymi modelami urządzeń danej klasy (np. różnymi joystickami). Pozwala to na przenośność i przejrzystość kodu.

### Hardware Emulation Layer (HEL)

Warstwa emulacji sprzętowej w systemach komputerowych. Sprowadza się ona do programowego emulowania działania pewnej funkcjonalności przez API obsługujące dany sprzęt, w wypadku gdy ów sprzęt sam w sobie takiej funkcjonalności nie posiada. Przykładem może być emulowanie teksturowania obiektów w wypadku gdy dany model karty graficznej sam w sobie nie potrafi teksturować obiektów.

### Human Interface Device (HID)

Urządzenia peryferyjne służące do wprowadzania danych do komputera przez człowieka i dawanie człowiekowi informacji zwrotnych przez komputer. Obecnie urządzenia HID są produkowane z myślą o podłączeniu ich do portu USB.