



# AGH

TECHNIKA MIKROPROCESOROWA

---

## Gesture Processing Library

---

*Autorzy:*

Michał JANIEC

Bartosz POLNIK

# Spis treści

|          |                                    |          |
|----------|------------------------------------|----------|
| <b>1</b> | <b>Temat</b>                       | <b>1</b> |
| <b>2</b> | <b>Cel</b>                         | <b>1</b> |
| <b>3</b> | <b>Opis zagadnienia</b>            | <b>1</b> |
| <b>4</b> | <b>Lista gestów</b>                | <b>1</b> |
| <b>5</b> | <b>Udostępniana funkcjonalność</b> | <b>4</b> |
| 5.1      | gp_Main.h . . . . .                | 4        |
| 5.1.1    | Gesture definitions . . . . .      | 4        |
| 5.1.2    | Data structures . . . . .          | 4        |
| 5.1.3    | Functions . . . . .                | 5        |
| 5.2      | gp_Alloc.h . . . . .               | 6        |
| 5.2.1    | Constants . . . . .                | 6        |
| 5.2.2    | Functions . . . . .                | 7        |
| 5.3      | gp.h . . . . .                     | 7        |
| 5.4      | gp_bool.h . . . . .                | 7        |
| 5.4.1    | Constants . . . . .                | 7        |
| 5.5      | gp_point.h . . . . .               | 7        |
| 5.5.1    | Data structures . . . . .          | 8        |
| 5.5.2    | Functions . . . . .                | 8        |
| 5.6      | gp_printf.h . . . . .              | 8        |
| 5.7      | gp_types.h . . . . .               | 8        |
| 5.7.1    | Data types . . . . .               | 9        |
| 5.7.2    | Constants . . . . .                | 9        |
| 5.8      | gp_vector.h . . . . .              | 9        |
| 5.8.1    | Data structures . . . . .          | 9        |
| 5.8.2    | Functions . . . . .                | 10       |
| 5.9      | gp_gestures_parameters.h . . . . . | 11       |
| 5.9.1    | Constants . . . . .                | 11       |
| 5.10     | gp_gestures_results.h . . . . .    | 11       |
| 5.10.1   | Constants . . . . .                | 12       |
| 5.11     | gp_MotionEvent.h . . . . .         | 12       |
| 5.11.1   | Constants . . . . .                | 12       |
| 5.12     | gp_OutputGesture.h . . . . .       | 12       |
| 5.12.1   | Data structures . . . . .          | 13       |
| 5.13     | gp_Math.h . . . . .                | 14       |
| 5.13.1   | Constants . . . . .                | 15       |

|                            |           |
|----------------------------|-----------|
| 5.13.2 Functions . . . . . | 15        |
| <b>6 Pliki</b>             | <b>19</b> |

# **1 Temat**

Stworzenie niskopoziomowej biblioteki do przetwarzania gestów, dedykowanej dla mikroprocesorów jedno-układowych.

## **2 Cel**







Celem projektu jest przede wszystkim stworzenie ww. biblioteki pozwalającej na wygodne korzystanie z technologii multi-touch na różnorodnych urządzeniach. Ponadto utworzona zostanie aplikacja na platformę Android służąca zaprezentowaniu działania biblioteki. Jej zadaniem będzie odczytywanie gestów wykonanych przez użytkownika i wyświetlanie ich nazw.




## **3 Opis zagadnienia**

Zadaniem biblioteki będzie odczytywanie gestów z urządzenia dotykowego. Biblioteka będzie okresowo odczytywać stan urządzenia (wejście biblioteki). Na tej podstawie będzie rozpoznawać ruchy, które będzie dopasowywać do listy gestów. Po rozpoznaniu gestu informacja o wykonanym geście zostanie umieszczona w kolejce gestów (wyjście biblioteki). Użytkownik biblioteki powinien okresowo sprawdzać czy coś pojawiło się w kolejce i samodzielnie przetwarzać jej zawartość. Gesty będą dodawane do biblioteki w czasie kompilacji. Biblioteka zostanie napisana w języku C bez użycia zewnętrznych bibliotek.

## **4 Lista gestów**

W celu uniknięcia niejednoznaczności proponujemy angielskie nazwy gestów.

| Nazwa      | Rysunek   | Opis   | Parametry                    |
|------------|---|--|------------------------------|
| Tap        |    | Pojedyncze stuknięcie w multi-touch.                 | Pozycja (x,y)                |
| Double Tap |    | Szybkie podwójne stuknięcie w multi-touch.           | Pozycja (x,y)                |
| Press      |    | Stuknięcie i przytrzymanie palca przez dłuższy czas. | Pozycja(x,y)                 |
| Move       |  | Przesunięcie palca w dowolnym kierunku.              | Pozycja(x,y)<br>Pozycja(x,y) |
| Rotate     |  | Obrót w lewo lub w prawo.                            | Left/Right<br>Obrót, (kąt)   |
| Flick      |  | Przesunięcie palca w lewo lub prawo i puszczenie.    | Left/Right,<br>Pozycja(x,y)  |

| Nazwa                            | Rysunek   | Opis  | Parametry                           |
|----------------------------------|---|---|-------------------------------------|
| <b>Scroll</b>                    |  | Przesunięcie palca w górę lub w dół i puszczenie.                 | Up/Down,<br>Pozycja(x,y)            |
| <b>Zoom</b>                      |  | Przybliżenie lub oddalenie palca wskazującego i kciuka do siebie. | In/Out,<br>Przybliżenie<br>(liczba) |
| <b>Two<br/>Finger<br/>Scroll</b> |  | Przesunięcie dwóch palców równolegle w górę lub w dół.            | Up/Down,<br>Pozycja(x,y)            |

## 5 Udostępniana funkcjonalność

### 5.1 gp\_Main.h

Udostępnia podstawowe funkcje realizowane przez bibliotekę.

#### 5.1.1 Gesture definitions

Zmienne zawierają szczegółowe informacje o wykonanym ruchu

- *gpOutputGesture\_tap* gp\_TapData
- *gpOutputGesture\_press* gp\_PressData
- *gpOutputGesture\_flick* gp\_FlickData
- *gpOutputGesture\_move* gp\_MoveData
- *gpOutputGesture\_rotation* gp\_RotationData
- *gpOutputGesture\_scroll* gp\_ScrollData
- *gpOutputGesture\_zoom* gp\_ZoomData
- *gpOutputGesture\_two\_finger\_scroll* gp\_TwoFingerScrollData
- *gpOutputGesture\_two\_finger\_tap* gp\_TwoFingerTapData

#### 5.1.2 Data structures

*gpRecognizeContext*

Przechowuje informacje o aktualnie wykonywanym ruchu

- *gpVector\* finger1* przechowuje zbiór współrzędnych wykonywanego ruchu dla pierwszego palca
- *gpVector\* finger2* przechowuje zbiór współrzędnych wykonywanego ruchu dla drugiego palca
- *gpByte fingers* przechowuje ilość palców biorących udział w ruchu
- *gpInt firstTime* przechowuje czas pierwszego dotknięcia ekranu w aktualnym ruchu

### 5.1.3 Functions

`gpVoid gpRecognize(gpMotionEvent* event)`

Odpowiada za rozpoznanie gestu

- *gpMotionEvent\* event* event do przetworzenia

`gpBool gpTryTap(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to tap

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryPress(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to press

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryFlick(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to flick

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryRotation(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to rotation

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryScroll(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to scroll

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryZoom(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to zoom

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryTwoFingerScroll(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to two finger scroll

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

`gpBool gpTryTwoFingerTap(gpMotionEvent* event, gpRecognizeContext* context)`

Sprawdza, czy aktualnie skończony ruch to two finger tap

- *gpMotionEvent\* event* ostatni otrzymany event związany z ruchem
- *gpRecognizeContext\* context* kontekst związany z ruchem

Warto zauważyć brak funkcji sprawdzającej, czy aktualnie skończony ruch to move. Wynika to z faktu, iż rozpoznawanie takiego ruchu następuje na bieżąco i informacja o tym ruchu dostępna jest nie tylko po otrzymaniu akcji dotyczącej oderwania ostatniego palca.

## 5.2 gp\_Alloc.h

Odpowiada za zarządzanie pamięcią

### 5.2.1 Constants

- *gpAlloc\_MAX\_MEM* 1000000 przechowuje rozmiar bufora pamięci tymczasowej



### 5.2.2 Functions

`gpVoid* gpAlloc_alloc(gpInt size)`

Przydziela pamięć z bufora

- *gpInt size* ilość jednostek pamięci do przydzielenia

`gpVoid gpAlloc_free(gpVoid* ptr)`

Zwalnia poprzednio przydzieloną z bufora pamięć

- *gpVoid\* ptr* wskaźnik na początek zaalokowanej poprzednio pamięci

`gpVoid gpAlloc_copy(gpVoid* from, gpVoid* to, gpInt size)`

Przekopiuje dane pomiędzy from do to. Nie przydziela pamięci.

- *gpVoid\* from* źródło danych do przeniesienia
- *gpVoid\* to* lokacja do której dane mają być przeniesione
- *gpInt size* ilość jednostek do przeniesienia

## 5.3 gp.h

Zawiera użyteczne include'y plików nagłówkowych z folderu base

## 5.4 gp\_bool.h

Definiuje podstawowe aliasy na wartości typu logicznego

### 5.4.1 Constants

- *false* 0
- *true* 1

## 5.5 gp\_point.h

Zawiera funkcje i struktury dotyczące punktu

### 5.5.1 Data structures

`gpPoint`

Reprezentacja punktu

- *gpFloat x* współrzędna x
- *gpFloat y* współrzędna y

### 5.5.2 Functions

`gpFloat gpPoint_distance(gpPoint* a, gpPoint* b)`

Oblicza dystans pomiędzy dwoma punktami w przestrzeni

- *gpPoint\* a* wskaźnik na pierwszy punkt
- *gpPoint\* b* wskaźnik na drugi punkt

`gpFloat gpPoint_distance2(gpPoint* a, gpPoint* b)`

Oblicza kwadrat odległości pomiędzy punktami

- *gpPoint\* a* wskaźnik na pierwszy punkt
- *gpPoint\* b* wskaźnik na drugi punkt

`gpPoint gpPoint_init(gpFloat x, gpFloat y)`

Tworzy punkt o zadanych współrzędnych

- *gpFloat x* współrzędna x
- *gpFloat y* współrzędna y

## 5.6 gp\_printf.h

Zawiera funkcje pomocne przy debugowaniu na platformie android

## 5.7 gp\_types.h

Definiuje abstrakcję na typy zależne od platformy

### 5.7.1 Data types

- *typedef void gpVoid*
- *typedef char gpBool*
- *typedef unsigned char gpUByte*
- *typedef signed char gpByte*
- *typedef unsigned short gpUWord*
- *typedef signed short gpWord*
- *typedef unsigned long gpUInt*
- *typedef signed long gpInt*
- *typedef char gpChar*
- *typedef char\* gpString*
- *typedef long gpFloat*

### 5.7.2 Constants

- *null*  $((gpVoid*)(0))$  pomocna reprezentacja wskaźnika o wartości nieokreślonej

## 5.8 gp\_vector.h

Opisuje abstrakcyjną kolekcję i operacje na niej

### 5.8.1 Data structures

*gpVector*  
kolekcja

- *gpVoid\*\* data* wskaźnik na pierwszy element kolekcji
- *gpInt capacity* aktualna pojemność kolekcji
- *gpInt size* aktualny rozmiar kolekcji

### 5.8.2 Functions

`gpVoid gpVector_init(gpVector* self)`

Inicjalizuje kolekcję (alokuje pamięć na jej elementy)

- *gpVector\* self* wskaźnik na zaalokowaną kolekcję

`gpVoid gpVector_destroy(gpVector* self)`

Zwalnia pamięć zajmowaną przez kolekcję

- *gpVector\* self* wskaźnik na kolekcję

`gplnt gpVector_getSize(gpVector* self)`

Zwraca rozmiar kolekcji

- *gpVector\* self* wskaźnik na kolekcję

`gpVoid* gpVector_at(gpVector* self, gplnt index)`

Zwraca wskaźnik na element kolekcji

- *gpVector\* self* wskaźnik na kolekcję
- *gplnt index* numer wskaźnika do elementu do zwrócenia

`gpVoid gpVector_clean(gpVector* self)`

Czyści zawartość kolekcji

- *gpVector\* self* wskaźnik na kolekcję

`gpVoid gpVector_pushBack(gpVector* self, gpVoid* what, gplnt size)` Dokłada element na koniec kolekcji

- *gpVector\* self* wskaźnik na kolekcję
- *gpVoid\* what* wskaźnik na element do dołożenia
- *gplnt size* rozmiar elementu dokładanego

`gpVoid gpVector_popBack(gpVector* self, gpVoid* where, gplnt size)`

- *gpVector\* self* wskaźnik na kolekcję

- *gpVoid\* where* wskaźnik na miejsce w pamięci, w które ma być przeniesiony ostatni element kolekcji
- *gpInt size* rozmiar elementu kolekcji

## 5.9 gp\_gestures\_parameters.h

Opisuje parametry dotyczące tolerancyjności w wykrywaniu ruchów

### 5.9.1 Constants

- *GP\_TAP\_MAX\_TIME* 40 maksymalny czas tap'a
- *GP\_TAP\_MAX\_MOVE* *gpMkFloat("12")* maksymalne odchylenie ruchu dla tapa
- *GP\_ROTATION\_MAX\_MOVE* *gpMkFloat("40")* maksymalne odchylenie podczas wykonywania rotacji
- *GP\_TAP\_PRESS\_MOVE* *gpMkFloat("10")* maksymalne odchylenie dla długiego przyciśnięcia
- *GP\_SCROLL\_MIN\_LEN* *gpMkFloat("20")* minimalna odległość dla ruchu scroll
- *GP\_FLICK\_MIN\_LEN* *gpMkFloat("15")* minimalna odległość dla ruchu flick
- *GP\_TWO\_FINGER\_TAP\_MAX\_DIST* *gpMkFloat("60")* maksymalna odległość dla two finger tap
- *GP\_ZOOM\_MIN\_CHANGE* *gpMkFloat("10")* minimalna odległość dla zoom

## 5.10 gp\_gestures\_results.h

Definiuje stałe pomocne przy rozpoznawaniu kierunków ruchu

### 5.10.1 Constants

- *GP\_SCROLL\_DOWN true* definiuje kierunek w dół
- *GP\_SCROLL\_UP false* definiuje kierunek w górę
- *GP\_FLICK\_LEFT false* definiuje ruch w lewo
- *GP\_FLICK\_RIGHT true* definiuje ruch w prawo
- *GP\_ZOOM\_IN true* definiuje przybliżenie
- *GP\_ZOOM\_OUT false* definiuje oddalenie

## 5.11 gp\_MotionEvent.h

Określa znacznie kodu akcji w ewencie

### 5.11.1 Constants

- *GP\_ME\_ACTION\_DOWN 0* początek ruchu
- *GP\_ME\_ACTION\_MOVE 2* ruch po ekranie
- *GP\_ME\_ACTION\_POINTER\_1\_DOWN 5* opuszczenie pierwszego palca na ekran
- *GP\_ME\_ACTION\_POINTER\_1\_UP 6* podniesienie pierwszego palca
- *GP\_ME\_ACTION\_POINTER\_2\_DOWN 261* opuszczenie drugiego palca
- *GP\_ME\_ACTION\_POINTER\_2\_UP 262* podniesienie drugiego palca
- *GP\_ME\_ACTION\_UP 1* koniec ruchu

## 5.12 gp\_OutputGesture.h

Zawiera struktury reprezentujące dodatkowe informacje o ruchu

### 5.12.1 Data structures

`gpOutputGesture_two_finger_scroll`

Reprezentuje two finger scroll

- *gpFloat x* przesuwanie poziome
- *gpFloat y* przesunięcie pionowe
- *gpBool direction* kierunek przewinięcia

`gpOutputGesture_zoom`

Reprezentuje zoom

- *gpBool direction* kierunek
- *gpFloat magnification* stosunek odległości

`gpOutputGesture_scroll`

Reprezentuje scroll

- *gpFloat x* przesuwanie poziome
- *gpFloat y* przesunięcie pionowe
- *gpBool direction* kierunek przewinięcia

`gpOutputGesture_flick`

Reprezentuje flick

- *gpFloat x* przemieszczenie poziome
- *gpFloat y* przemieszczenie pionowe
- *gpBool direction* kierunek przemieszczenia

`gpOutputGesture_rotation`

Reprezentuje rotation

- *gpBool direction* informacja, czy kąt jest dodatni, czy ujemny
- *gpFloat angle* kąt obrotu

#### `gpOutputGesture_move`

Reprezentuje przesunięcie

- *gpFloat x* początkowa współrzędna x
- *gpFloat y* początkowa współrzędna y
- *gpFloat begx* końcowa współrzędna x
- *gpFloat begy* końcowa współrzędna y

#### `gpOutputGesture_press`

Reprezentuje press

- *gpFloat x* współrzędna x
- *gpFloat y* współrzędna y

#### `gpOutputGesture_tap`

Reprezentuje tap

- *gpFloat x* współrzędna x
- *gpFloat y* współrzędna y

#### `gpOutputGesture_two_finger_tap`

Reprezentuje two finger tap

- *gpFloat x* współrzędna x
- *gpFloat y* współrzędna y

### **5.13 gp\_Math.h**

Zawiera deklarację zaimplementowanych funkcji oraz stałych



### 5.13.1 Constants

- *GP\_FLOAT\_BASE* 10000 część liczby przeznaczona na miejsca dziesiętne
- *gpMath\_EPSILON* 10 epsilon
- *gpMath\_PI* 31416 PI
- *gpMath\_2PI* 62832 dwukrotność PI
- *gpMath\_PI2* 15708 połowa liczby PI
- *gpMath\_PI4* 7854 PI przez 4
- *gpMath\_PI6* 5236 PI przez 6
- *gpMath\_E* 27183 E
- *gpMath\_1* 10000 1
- *gpMath\_2* 20000 2
- *gpMath\_3* 30000 3
- *gpMath\_SINPI4* 7071 sinus PI przez 4
- *gpMath\_0* 0 0

### 5.13.2 Functions

**gpFloat gpMul(gpFloat a, gpFloat b)**  
Mnoży dwie liczby stałopozycyjne

- *gpFloat a* mnożna
- *gpFloat b* mnożnik

**gpFloat gpDiv(gpFloat a, gpFloat b)**  
Dzieli dwie liczby stałopozycyjne

- *gpFloat a* dzielna
- *gpFloat b* dzielnik

`gpFloat gpSub(gpFloat a, gpFloat b)`  
odejmuje dwie liczby stałopozycyjne

- *gpFloat a*    odjemna
- *gpFloat b*    odjemnik

`gpFloat gpAdd(gpFloat a, gpFloat b)`  
Dodaje dwie liczby stałopozycyjne

- *gpFloat a*    składnik
- *gpFloat b*    składnik

`gpFloat gpNeg(gpFloat a)`  
Zwraca liczbę przeciwną do danej liczby stałopozycyjnej

- *gpFloat a*    operand

`gpInt gpMath_MinInt()`  
Zwraca mniejszą z liczb typu `gpInt`

- *gpFloat a*    operand
- *gpFloat b*    operand

`gpByte gpMath_Sign(gpFloat x)`  
Zwraca signum liczby

- *gpFloat x*    operand

`gpFloat gpMath_Abs(gpFloat a)`  
Zwraca wartość bezwzględną liczby

- *gpFloat a*    operand

`gpFloat gpMath_Square(gpFloat a)`  
Zwraca kwadrat liczby

- *gpFloat a* operand

`gpFloat gpMath_Sqrt(gpFloat a)`

Zwraca pierwiastek liczby

- *gpFloat a* operand

`gpFloat gpMath_Exp(gpFloat a)`

Zwraca  $E$  podniesione do potęgi  $a$

- *gpFloat a* operand

`gpFloat gpMath_Powl(gpFloat base, gpInt exp)`

Potęguje

- *gpFloat base* liczba do podniesienia
- *gpInt exp* wykładnik

`gpFloat gpMath_Sin(gpFloat x)`

Oblicza sinus

- *gpFloat x* kąt w radianach

`gpFloat gpMath_Cos(gpFloat x)`

Oblicza kosinus

- *gpFloat x* kąt podany w radianach

`gpFloat gpMath_Tan(gpFloat x)`

Oblicza tangens

- *gpFloat x* kąt podany w radianach

`gpFloat gpMath_ATan2(gpFloat x, gpFloat y)`

Oblicza wartość bliższego kąta pomiędzy punktem, a osią  $x$

- *gpFloat x* współrzędna  $x$
- *gpFloat y* współrzędna  $y$

`gpFloat gpMath_ASin(gpFloat x)`

Oblicza arcus-sinus

- *gpFloat x* kąt w radianach

`gpFloat gpMath_ACos(gpFloat x)`

Oblicza arcus-kosinus

- *gpFloat x* kąt podany w radianach

`gpFloat gpMath_ATan(gpFloat x)`

Oblicza arcus-tangens

- *gpFloat x* kąt podany w radianach

`gpFloat gpMath_MinFloat(gpFloat a, gpFloat b)`

Zwraca mniejszą z liczb

- *gpFloat a* operand
- *gpFloat b* operand

`gpFloat gpMath_MaxFloat(gpFloat a, gpFloat b)`

Zwraca większą z liczb

- *gpFloat a* operand
- *gpFloat b* operand

`gpBool gpMath_Equals(gpFloat a, gpFloat b)`

Porównuje dwie liczby

- *gpFloat a* operand
- *gpFloat b* operand

`gpInt gpMath_Int(gpFloat a)`

Zamienia liczbę stałopozycyjną na całkowitą

- *gpFloat a* liczba stałopozycyjna

**gpFloat gpMath\_FloatI(gpInt a)**

Zwraca liczbę całkowitą na stałopozycyjną

- *gpInt a* liczba całkowita

**gpFloat gpMath\_AngleToAzimut(gpPoint a, gpPoint b)**

Oblicza kąt skierowany między punktami

- *gpPoint a* pierwszy punkt
- *gpPoint b* drugi punkt

**gpFloat gpMkFloat(gpString x)**

Tworzy liczbę stałopozycyjną z napisu

- *gpString x* napis

## 6 Pliki

- *gp\_Main.h* project\_c\BaseProject\Include
- *gp\_Alloc.h* project\_c\BaseProject\Include\Alloc
- *gp.h* project\_c\BaseProject\Include\Base
- *gp\_bool.h* project\_c\BaseProject\Include\Base
- *gp\_point.h* project\_c\BaseProject\Include\Base
- *gp\_printf.h* project\_c\BaseProject\Include\Base
- *gp\_types.h* project\_c\BaseProject\Include\Base
- *gp\_vector.h* project\_c\BaseProject\Include\Base
- *gp\_gestures\_parameters.h* project\_c\BaseProject\Include\Gestures
- *gp\_gestures\_results.h* project\_c\BaseProject\Include\Gestures

- *gp\_MotionEvent.h* project\_c\BaseProject\Include\InOut
- *gp\_OutputGesture.h* project\_c\BaseProject\Include\InOut
- *gp\_Math.h* project\_c\BaseProject\Include\Math
- *gp\_Main.c* project\_c\BaseProject\Source
- *gp\_Alloc.c* project\_c\BaseProject\Source\Alloc
- *gp.c* project\_c\BaseProject\Source\Base
- *gp\_point.c* project\_c\BaseProject\Source\Base
- *gp\_printf.c* project\_c\BaseProject\Source\Base
- *gp\_vector.c* project\_c\BaseProject\Source\Base
- *gp\_flick.c* project\_c\BaseProject\Source\Gestures
- *gp\_gestures.c* project\_c\BaseProject\Source\Gestures
- *gp\_press.c* project\_c\BaseProject\Source\Gestures
- *gp\_rotation.c* project\_c\BaseProject\Source\Gestures
- *gp\_scroll.c* project\_c\BaseProject\Source\Gestures
- *gp\_tap.c* project\_c\BaseProject\Source\Gestures
- *gp\_zoom.c* project\_c\BaseProject\Source\Gestures
- *gp\_MotionEvent.c* project\_c\BaseProject\Source\InOut
- *gp\_Math.c* project\_c\BaseProject\Source\Math