



AGH

TECHNIKA MIKROPROCESOROWA

Gesture Processing Library

Autorzy:

Michał JANIEC

Bartosz POLNIK

Spis treści

1	Temat	2
2	Cel	2
3	Opis zagadnienia	2
4	Lista gestów	2
5	Udostępniana funkcjonalność	5
5.1	gp_Main.h	5
5.1.1	Gesture declarations	5
5.1.2	Data structures	5
5.1.3	Functions	5
5.2	gp_Alloc.h	7
5.2.1	Constants	7
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	7
5.5.2	Functions	8
5.6	gp_printf.h	8
5.6.1	Constants	8
5.7	gp_types.h	8
5.7.1	Data types	8
5.7.2	Constants	9
5.8	gp_vector.h	9
5.8.1	Data structures	9
5.8.2	Functions	9
5.9	gp_gestures_parameters.h	10
5.9.1	Constants	10
5.10	gp_gestures_results.h	11
5.10.1	Constants	11
5.11	gp_MotionEvent.h	11
5.11.1	Constants	11
5.12	gp_OutputGesture.h	12
5.12.1	Data structures	12
5.13	gp_Math.h	13

5.13.1 Constants	13
5.13.2 Functions	14
6 File location	17

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.




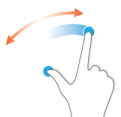

3 Opis zagadnienia




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)
Press		Stuknięcie i przytrzymanie palca przez dłuższy czas.	Pozycja(x,y)
Move		Przesunięcie palca w dowolnym kierunku.	Pozycja(x,y) Pozycja(x,y)
Rotate		Obrót w lewo lub w prawo.	Left/Right Obrót, (kąt)
Flick		Przesunięcie palca w lewo lub prawo i puszczenie.	Left/Right, Pozycja(x,y)

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

5 Udostępniana funkcjonalność

5.1 gp_Main.h

5.1.1 Gesture declarations

- *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

- *gpVector* finger1* opis
- *gpVector* finger2* opis
- *gpByte fingers* opis
- *gpInt firstTime* opis

5.1.3 Functions

gpVoid gpRecognize(gpMotionEvent* event)

- *gpMotionEvent* event* opis

gpBool gpTryTap(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryPress(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryFlick(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryRotation(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryScroll(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryZoom(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryTwoFingerScroll(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

gpBool gpTryTwoFingerTap(gpMotionEvent* event, gpRecognizeContext* context)

- *gpMotionEvent* event* opis
- *gpRecognizeContext* context* opis

5.2 gp_Alloc.h

5.2.1 Constants

- *gpAlloc_MAX_MEM* 1000000 opis

5.2.2 Functions

gpVoid* gpAlloc_alloc(gpInt size)

- *gpInt size* opis

gpVoid gpAlloc_free(gpVoid* ptr)

- *gpVoid* ptr* opis

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

- *gpVoid* from* opis
- *gpVoid* to* opis
- *gpInt size* opis

5.3 gp.h

5.4 gp_bool.h

5.4.1 Constants

- *false* 0 opis
- *true* 1 opis

5.5 gp_point.h

5.5.1 Data structures

gpPoint

- *gpFloat x* opis
- *gpFloat y* opis

5.5.2 Functions

gpFloat gpPoint_distance(gpPoint* a, gpPoint* b)

- *gpPoint* a* opis
- *gpPoint* b* opis

gpFloat gpPoint_distance2(gpPoint* a, gpPoint* b)

- *gpPoint* a* opis
- *gpPoint* b* opis

gpPoint gpPoint_init(gpFloat x, gpFloat y)

- *gpFloat x* opis
- *gpFloat y* opis

5.6 gp_printf.h

5.6.1 Constants

- *printf* *--android_printf* opis

5.7 gp_types.h

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))* opis

5.8 gp_vector.h

5.8.1 Data structures

gpVector

- *gpVoid** data* opis
- *gpInt capacity* opis
- *gpInt size* opis

5.8.2 Functions

gpVoid gpVector_init(gpVector* self)

- *gpVector* self* opis

gpVoid gpVector_destroy(gpVector* self)

- *gpVector* self* opis

gpInt gpVector_getSize(gpVector* self)

- *gpVector* self* opis

gpVoid* gpVector_at(gpVector* self, gpInt index)

- *gpVector* self* opis

- *gpInt index* opis

`gpVoid gpVector_clean(gpVector* sefl)`

- *gpVector* sefl* opis

`gpVoid gpVector_pushBack(gpVector* self, gpVoid* what, gpInt size)`

- *gpVector* self* opis

- *gpVoid* what* opis

- *gpInt size* opis

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

- *gpVector* self* opis

- *gpVoid* where* opis

- *gpInt size* opis

5.9 gp_gestures_parameters.h

5.9.1 Constants

- *GP_DEBOUNCE_TIME* 5 opis
- *GP_TAP_MAX_TIME* 40 opis
- *GP_TAP_MAX_MOVE* *gpMkFloat("12")* opis
- *GP_ROTATION_MAX_MOVE* *gpMkFloat("20")* opis
- *GP_TAP_PRESS_MOVE* *gpMkFloat("10")* opis
- *GP_SCROLL_MIN_LEN* *gpMkFloat("20")* opis
- *GP_FLICK_MIN_LEN* *gpMkFloat("15")* opis
- *GP_TWO_FINGER_TAP_MAX_DIST* *gpMkFloat("60")* opis
- *GP_ZOOM_MIN_CHANGE* *gpMkFloat("10")* opis

5.10 gp_gestures_results.h

5.10.1 Constants

- *GP_SCROLL_DOWN* *true* opis
- *GP_SCROLL_UP* *false* opis
- *GP_FLICK_LEFT* *false* opis
- *GP_FLICK_RIGHT* *true* opis
- *GP_ZOOM_IN* *true* opis
- *GP_ZOOM_OUT* *false* opis

5.11 gp_MotionEvent.h

5.11.1 Constants

- *GP_ME_ACTION_CANCEL* *3* opis
- *GP_ME_ACTION_DOWN* *0* opis
- *GP_ME_ACTION_HOVER_ENTER* *9* opis
- *GP_ME_ACTION_HOVER_EXIT* *10* opis
- *GP_ME_ACTION_HOVER_MOVE* *7* opis
- *GP_ME_ACTION_MOVE* *2* opis
- *GP_ME_ACTION_POINTER_1_DOWN* *5* opis
- *GP_ME_ACTION_POINTER_1_UP* *6* opis
- *GP_ME_ACTION_POINTER_2_DOWN* *261* opis
- *GP_ME_ACTION_POINTER_2_UP* *262* opis
- *GP_ME_ACTION_POINTER_DOWN* *5* opis
- *GP_ME_ACTION_POINTER_UP* *6* opis
- *GP_ME_ACTION_SCROLL* *8* opis
- *GP_ME_ACTION_UP* *1* opis

5.12 gp_OutputGesture.h

5.12.1 Data structures

gpOutputGesture_two_finger_scroll

- *gpFloat x* opis
- *gpFloat y* opis
- *gpBool direction* opis

gpOutputGesture_zoom

- *gpBool direction* opis
- *gpFloat magnification* opis

gpOutputGesture_scroll

- *gpFloat x* opis
- *gpFloat y* opis
- *gpBool direction* opis

gpOutputGesture_flick

- *gpFloat x* opis
- *gpFloat y* opis
- *gpBool direction* opis

gpOutputGesture_rotation

- *gpBool direction* opis
- *gpFloat angle* opis

gpOutputGesture_move

- *gpFloat x* opis

- *gpFloat y* opis
- *gpFloat begx* opis
- *gpFloat begy* opis

gpOutputGesture_press

- *gpFloat x* opis
- *gpFloat y* opis

gpOutputGesture_tap

- *gpFloat x* opis
- *gpFloat y* opis

gpOutputGesture_two_finger_tap

- *gpFloat x* opis
- *gpFloat y* opis

5.13 gp_Math.h

5.13.1 Constants

- *GP_FLOAT_BASE* 10000 opis
- *gpMath_EPSILON* 10 opis
- *gpMath_PI* 31416 opis
- *gpMath_2PI* 62832 opis
- *gpMath_PI2* 15708 opis
- *gpMath_PI4* 7854 opis
- *gpMath_PI6* 5236 opis

- *gpMath_E* 27183 opis
- *gpMath_1* 10000 opis
- *gpMath_2* 20000 opis
- *gpMath_3* 30000 opis
- *gpMath_SINPI4* 7071 opis
- *gpMath_0* 0 opis

5.13.2 Functions

`gpFloat gpMul(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpFloat gpDiv(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpFloat gpSub(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpFloat gpAdd(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpFloat gpNeg(gpFloat a)`

- *gpFloat a* opis

gpInt gpMath_MinInt()

gpByte gpMath_Sign(gpFloat x)

- *gpFloat x* opis

gpFloat gpMath_Abs(gpFloat a)

- *gpFloat a* opis

gpFloat gpMath_Square(gpFloat a)

- *gpFloat a* opis

gpFloat gpMath_Sqrt(gpFloat a)

- *gpFloat a* opis

gpFloat gpMath_Exp(gpFloat a)

- *gpFloat a* opis

gpFloat gpMath_Powl(gpFloat base, gpInt exp)

- *gpFloat base* opis
- *gpInt exp* opis

gpFloat gpMath_Sin(gpFloat x)

- *gpFloat x* opis

gpFloat gpMath_Cos(gpFloat x)

- *gpFloat x* opis

gpFloat gpMath_Tan(gpFloat x)

- *gpFloat x* opis

`gpFloat gpMath_ATan2(gpFloat x, gpFloat y)`

- *gpFloat x* opis
- *gpFloat y* opis

`gpFloat gpMath_ASin(gpFloat x)`

- *gpFloat x* opis

`gpFloat gpMath_ACos(gpFloat x)`

- *gpFloat x* opis

`gpFloat gpMath_ATan(gpFloat x)`

- *gpFloat x* opis

`gpFloat gpMath_MinFloat(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpFloat gpMath_MaxFloat(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpBool gpMath_Equals(gpFloat a, gpFloat b)`

- *gpFloat a* opis
- *gpFloat b* opis

`gpInt gpMath_Int(gpFloat a)`

- *gpFloat a* opis

`gpFloat gpMath_Floatl(gpInt a)`

- *gpInt a* opis

`gpFloat gpMath_AngleToAzimut(gpPoint a, gpPoint b)`

- *gpPoint a* opis
- *gpPoint b* opis

`gpFloat gpMkFloat(gpString x)`

- *gpString x* opis

6 File location

- *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