



*Optyczny czujnik przemieszczenia.*

Edgar Ostrowski

Jan Kędzierski

Wrocław, 19.11.2006

## Spis treści

1	Wstęp.....	3
2	Zasada działania czujników.....	3
3	Przegląd czujników optycznych.....	4
4	Agilet ADNS-2051 .....	6
4.1	Opis układu .....	6
4.2	Opis wyjść kwadraturowych .....	7
4.3	Zapisywanie danych .....	7
4.4	Czytanie danych .....	8
4.5	Rejestry ADNS-2051 .....	9
5	Schemat montażu .....	18
6	Oprogramowanie .....	19
7	Realizacja projektu .....	23
8	Literatura .....	25

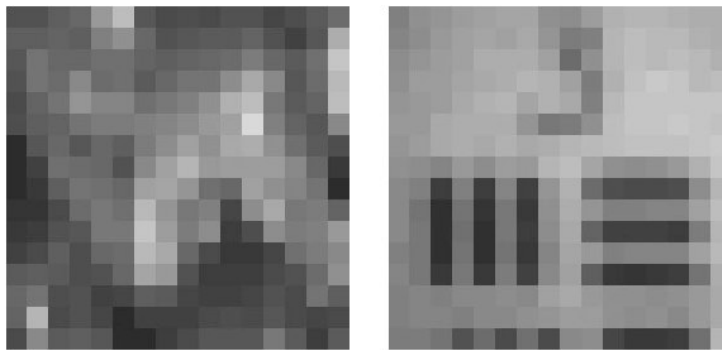
# 1 Wstęp

Niniejszy raport jest pierwszym sprawozdaniem okresowym z zajęć projektowych „Systemy Autonomiczne i Roboty Mobilni ” prowadzone przez dr Andrzeja Wołczowskiego w semestrze zimowym, roku akademickiego 2006/07 na Wydziale Elektroniki, Politechniki Wrocławskiej.

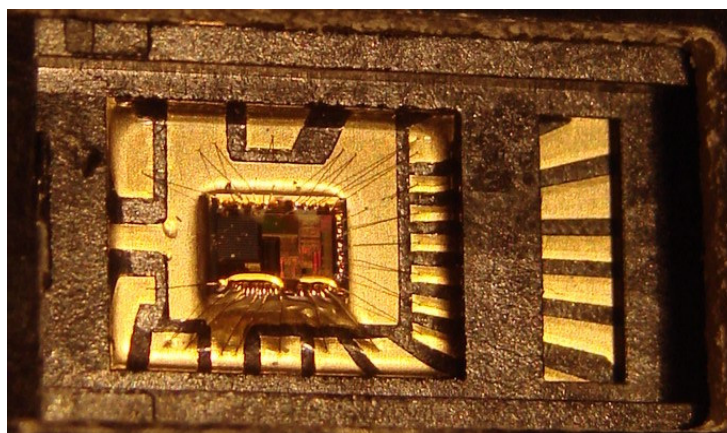
Dokument zawiera przegląd czujników stosowanych w popularnych myszkach optycznych. Na konkretnym przykładzie (Agilent ADNS-2051) przedstawiono szczegółowe parametry, sposób komunikacji oraz montaż układu.

## 2 Zasada działania czujników

Zasada działania czujników znajdujących się w myszkach optycznych jest oparta na Technologii Nawigacji Optycznej (Optical Navigation Technology). Wewnątrz układów znajduje się miniaturowy monochromatyczny aparat cyfrowy, który „fotografuje” nawierzchnie pod myszką. Konieczne też jest stosowanie odpowiedniego układu optycznego oraz dodatkowego źródła światła w postaci diody LED. Na podstawie zmieniającego się obrazu, układ oblicza zmianę położenia w pionie i poziomie.



Rysunek 1 - Przykłady odczytanych obrazów.



Rysunek 2 - Wewnętrzna budowa układu.

### 3 Przegląd czujników optycznych

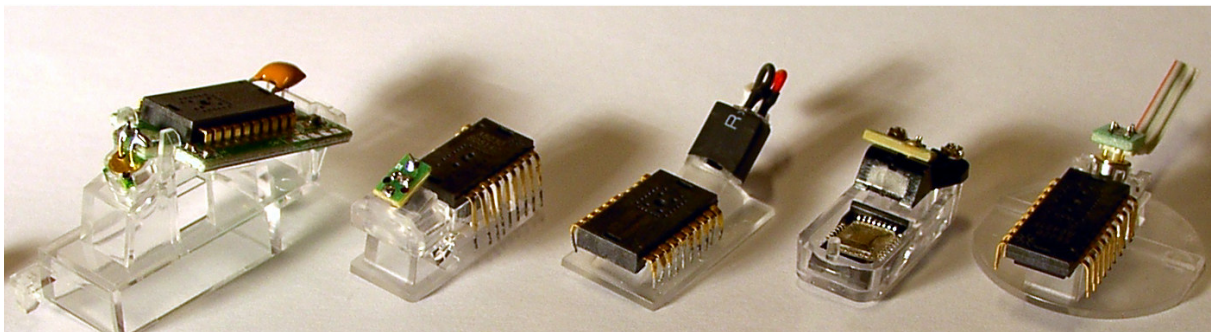
Zdecydowana większość myszek, która jest odstępna na rynku komputerowym zawiera układy produkowane przez następujące firmy:

- Agilent – [www.agilent.com](http://www.agilent.com)
- Avago – [www.avagotech.com](http://www.avagotech.com)
- ST Microelectronics – [www.st.com](http://www.st.com)
- Elan Microelectronics – [www.emc.com.tw](http://www.emc.com.tw)
- Pixart imaging – [www.pixar.com.tw](http://www.pixar.com.tw)
- AtLab Inc. – [www.atl21c.com](http://www.atl21c.com)

Czujniki można porównywać za pomocą następujących parametrów:

- **Wielkość matrycy** – ilość pikseli, które posiada matryca układu. Większa matryca umożliwi wykonywanie szybszych ruchów.
- **Rozdzielczość** – zazwyczaj podawana w jednostkach CPI (counts per inch) lub DPI (dots per inch), określa czułość układu na zmiany. Większa rozdzielczość zapewnia większą dokładność.
- **Odświeżanie (Hz)** – częstotliwością z jaką „fotografowana” jest nawierzchnia. Określa ile może zmienić się położenie bez utraty informacji.
- **Moc obliczeniowa (Mpixels/sec)** – jest to iloczyn ilości pikseli w matrycy i ilość zdjęć wykonywanych na sekundę. Układy o różnych parametrach mogą mieć taką samą moc obliczeniową i zwracać porównywalne wyniki. Oczywiście jest, że większa moc obliczeniowa zapewnia lepsze rezultaty.
- **Maksymalna prędkość (mm/sec)** – prędkość przy której można się poruszać bez utraty informacji o zmianie przemieszczenia. W przypadku gdy ten współczynnik jest za niski, układ generuje przypadkowe ruchy.
- **Maksymalne przyspieszenie (g)** – określa jak szybko można zmienić kierunek ruchu. O wielkości tego parametru decyduje zaimplementowany przez producenta algorytm.

Tabela nr 1 przedstawia przegląd czujników dostępnych w popularnych myszkach optycznych [1][2][3].



Rysunek nr 3 – Różne czujniki optyczne.

Nazwa układu	Myszki w których można znaleźć układ	Wielkość matrycy	Rozdzielczość (CPI)	Odświeżanie (Hz)	Mpixel/sec	Max prędkość (mm/sec)	Max przyspieszenie (g)	Uwagi:
Agilent HDNS-2000	Apple Pro, MS IntelliMouse Optical, MS Wheel Mouse Optical	18x18	400	1500	0.486	304	0.15	16 nóżek, obudowa DIL, rzadko spotykany
Avago/Agilent ADNS-2051	Logitech M-BJ58, Logitech M-BJ69, Logitech MouseMan, Apple Pro,	16x16	400 lub 800	od 1500 do 2300	0.588	355	0.15	16 nóżek, obudowa DIL, wyjścia kwadraturowe, ogólnie dostępny
Avago/Agilent ADNS-2030	ViewSonic MW407, Wireless Mouse	16x16	400 lub 800	2300	0.588	355	0.15	16 nóżek, obudowa DIL, rzadko spotykany
Avago/Agilent ADNS-2610	Logitech M-BJ90, ViewSonic CC2201, ViewSonic CP1204	18x18	400	1500	0.486	304	0.25	8 nóżek, obudowa DIL, ogólnie dostępny
ST Microelectronics 9N MLT 03	MS IntelliMouse Explorer	22x22	400	6000	2.904	?	?	44 nóżek, układ dostępny tylko dla firmy Microsoft
ST Microelectronics "Aviator"	MS Wireless IntelliMouse Explorer, MS Wireless Optical Mouse	22x22	400	6000	2.904	914	?	32 nóżek, układ dostępny tylko dla firmy Microsoft

Tabela nr 1 – Przegląd czujników

## 4 Agilet ADNS-2051

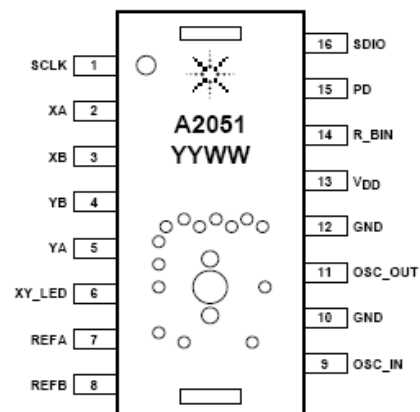
### 4.1 Opis układu

Zdecydowano wykorzystać układ ADNS-2051 firmy Agilent. O wyborze tego układu zdecydowały następujące kryteria:

- ogólnie dostępna dokumentacja układu [1],
- napięcie zasilania 5 V,
- znany sposób komunikacji (SDIO),
- analogowe wyjścia kwadraturowe (tzn. można traktować jak zwykłe enkodery),
- wysoka rozdzielczość 800 CPI,
- wbudowany sterownik diody LED, która oświetla nawierzchnie,
- niska cena myszki, w której znajduje się układ (Logitech M-BJ58).

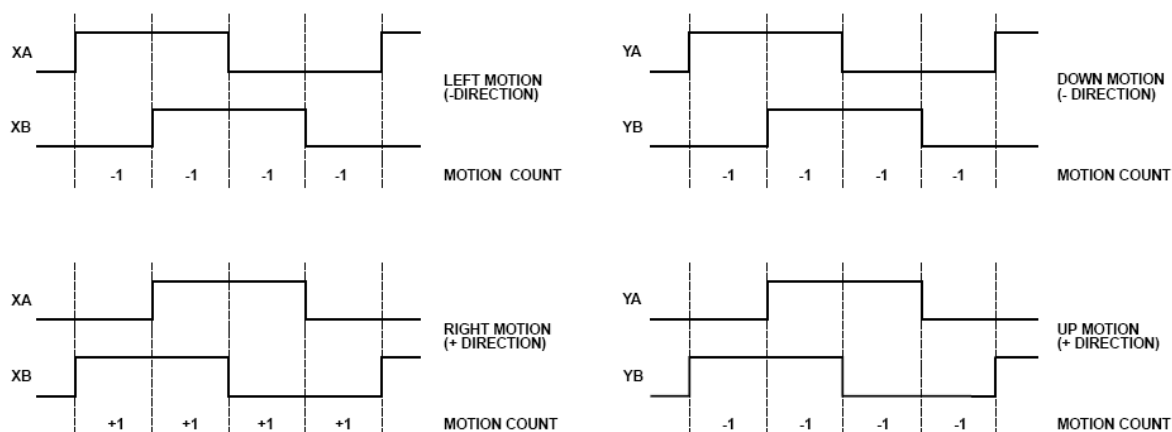
Pin nr	Nazwa sygnału	Opis
1	SCLK	Sygnal zegarowy portu szeregowego
2	XA	Wyjście kwadraturowe
3	XB	Wyjście kwadraturowe
4	YA	Wyjście kwadraturowe
5	YB	Wyjście kwadraturowe
6	XY_LED	Kontrola diody LED
7	REFA	Napięcie referencyjne
8	REFB	Napięcie referencyjne
9	OSC_IN	Wejście oscylatora
10	GND	Masa
11	OSC_OUT	Wyjście oscylatora
12	GND	Masa
13	V <sub>DD</sub>	Napięcie zasilania 5 V
14	R_BIN	Ograniczenie prądu dla diody
15	PD	Sygnalizacja pracy
16	SDIO	Port szeregowy (we/wy)

Tabela nr 2 – Opis wyprowadzeń układu.



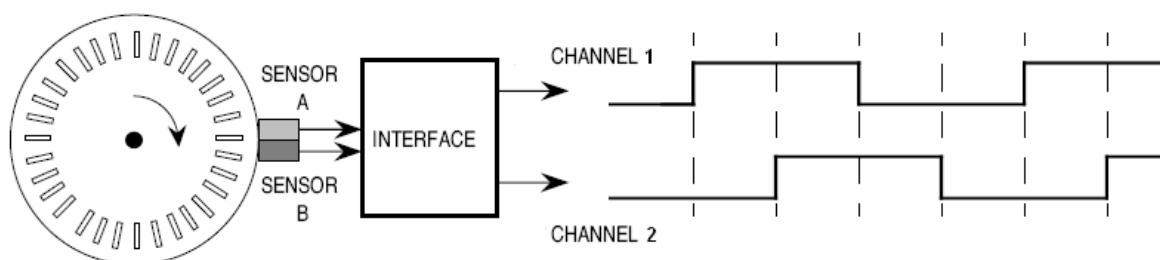
Rysunek nr 4 – Schemat układu.

## 4.2 Opis wyjść kwadraturowych



Rysunek nr 5 - Przebiegi na wyjściach kwadraturowych opisywanego czujnika.

Korzystając z wyjść kwadraturowych można użyć czujnika jako zwykły enkoder (rysunek 6).



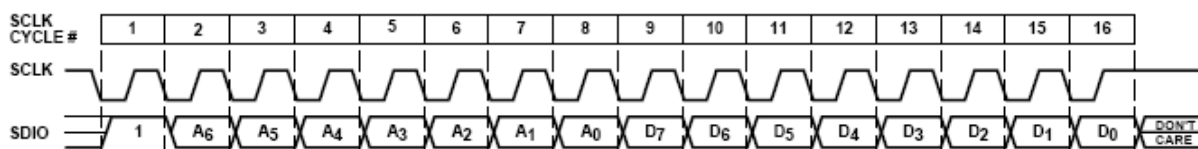
Rysunek nr 6 - Przebiegi na wyjściach kwadraturowych klasycznego enkodera.

Przy czym warto również skorzystać z opcji które opisano poniżej. Zapisu i odczytu parametrów dokonuje się poprzez interfejs szeregowy opisany w dalszej części raportu.

## 4.3 Zapisywanie danych

Komunikacja z układem odbywa się poprzez synchroniczny port szeregowy. Do tego celu wykorzystuje się linie: **SCLK** – sygnał zegarowy generowany przez mastera (mikrokontroler) oraz **SDIO** – dwukierunkowa linia danych. Możliwe jest programowanie oraz czytanie rejestrów układu.

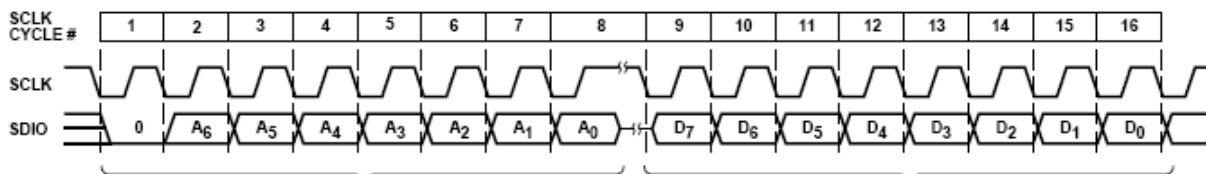
Zapisywanie informacji do układu jest zawsze inicjowane przez mastera i składa się z dwóch bajtów. Pierwszy bajt zawiera 7-bitowy adres rejestru do, którego mają zostać zapisane dane oraz „1” na najstarszym bicie aby wskazać kierunek przepływu danych. Drugi bajt zawiera dane, które mają zostać wpisane. Transfer jest synchronizowany przez sygnał SCLK. Master zmienia stan SDIO na opadających zboczach SCLK, układ czyta informacje na narastających zboczach SCLK.



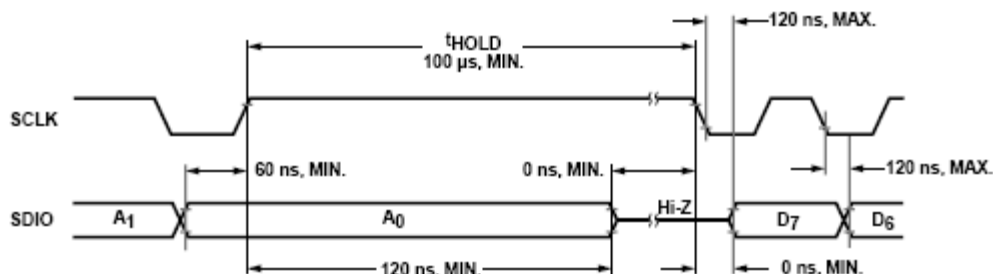
Rysunek 7 – Przebieg czasowy operacji zapisu.

#### 4.4 Czytanie danych

Operację czytania danych z układu inicjuje master. Składa się z dwóch bajtów. Pierwszy bajt zawiera 7-bitowy adres wysyłany przez mastera oraz „0” na najstarszym bicie, które wskazuje kierunek przepływu informacji. Drugi bajt zawiera informacje wysyłane przez układ. Transfer synchronizowany jest przez sygnał zegarowy SCLK. Zmiana SDIO odbywa się na opadających zboczach SCLK a czytanie na narastających zboczach SCLK. Mikrokontroler po wysłaniu ostatniego bitu adresu musi przejść w stan wysokiej impedancji a SCLK musi odczekać minimum 100  $\mu$ s. Jest to konieczne ponieważ układ potrzebuje czas na przygotowanie danych.



Rysunek 8 – Przebieg czasowy operacji odczytu.



Rysunek 9 – Czekanie na odbiór danych.



## 4.5 Rejestry ADNS-2051

- Tabela nr 3 przedstawia wszystkie dostępne rejestry ADNS-2051.

Adres	Nazwa rejestru	Opis
0x00	Produkt_ID	Identyfikator produktu, wartość stała.
0x01	Revition_ID	Wersja produktu.
0x02	Motion	Informuje czy nastąpił ruch.
0x03	Delta_X	Zmiana położenia w poziomie. Czytanie zeruje wartość.
0x04	Delta_Y	Zmiana położenia w pionie. Czytanie zeruje wartość.
0x05	SQUAL	Informuje o jakości powierzchni.
0x06	Average_Pixel	Średnia wartość wszystkich pikseli.
0x07	Maximum_Pixel	Najjaśniejszy piksel.
0x08	Reserved	Brak dostępu.
0x09	Reserved	Brak dostępu.
0x0a	Configuration_bits	Bity konfiguracyjne.
0x0b	Reserved	Brak dostępu.
0x0c	Data_Out_Lower	Rejestr uniwersalny.
0x0d	Data_Out_Upper	Rejestr uniwersalny.
0x0e	Shutter_Lower	Rejestr czasu migawki.
0x0f	Shutter_Upper	-  -
0x10	Frame_Period_Lower	Liczba klatek na sekundę.
0x11	Frame_Period_Upper	-  -

Tabela nr 3 – Spis rejestrów ADNS-2051

- Produkt\_ID** rejestr **0x00** dostęp: **tylko odczyt**

Identyfikator produktu, wartość stała.

7	6	5	4	3	2	1	0
PID <sub>7</sub>	PID <sub>6</sub>	PID <sub>5</sub>	PID <sub>4</sub>	PID <sub>3</sub>	PID <sub>2</sub>	PID <sub>1</sub>	PID <sub>0</sub>

Jest to 8-bitowy rejestr, który można tylko odczytywać. Wykorzystuje się go między innymi do sprawdzenia komunikacji z czujnikiem.

- Revition\_ID** rejestr **0x01** dostęp: **tylko odczyt**

Wersja produktu.

7	6	5	4	3	2	1	0
RID <sub>7</sub>	RID <sub>6</sub>	RID <sub>5</sub>	RID <sub>4</sub>	RID <sub>3</sub>	RID <sub>2</sub>	RID <sub>1</sub>	RID <sub>0</sub>

Jest to 8-bitowy rejestr, który można również tylko odczytywać. Zawiera informacje o wersji produktu. Wartości w nim zawarte mogą znajdować się pomiędzy 0x00-0xFF.

- Motion** rejestr **0x02** dostęp: **tylko odczyt**

Informuje czy nastąpił ruch.

7	6	5	4	3	2	1	0
MOT	Reserved	FAULT	OVFY	OVFX	Reserved	Reserved	RES

8-bitowy rejestr, który można również tylko odczytywać. Zawiera informacje o tym czy wystąpiło przemieszczenie czujnika od ostatniego odczytu. Jeśli tak użytkownik powinien odczytać rejestry 0x03 oraz 0x04. Można również pobrać informacje o tym czy bufor delta i deltaY zostały przekroczone. W rejestrze tym znajduje się również informacja o używanej aktualnie rozdzielczości.

### MOT

- 0 – nie wystąpił ruch.
- 1- ruch wystąpił od ostatniego odczytania deltaX deltaY.

### FAULT

- 0 – nie wystąpił błąd diody.
- 1 – prąd diody za wysoki lub za niski

### OVFY

- 0 – rejestr deltaY nie przepelniony
- 1- rejestr przepelniony

### OVFX

- 0 – rejestr deltaX nie przepelniony
- 1- rejestr przepelniony

### RES

- 0 – rozdzielczość 400dpi
- 1- 800dpi

Odczytanie tego rejestru powoduje zablokowanie na ten czas inkrementacji rejestrów deltaX oraz deltaY. Powinno się go odczytywać tuż przed odczytaniem delta\_X oraz delta\_Y. Jeśli odczytamy rejestr ponownie nie odczytując rejestrów delta\_X oraz delta\_Y to wartości w nich zgromadzone zostaną utracone.

Zaleca się więc czytanie rejestrów 0x02, 0x03, 0x04 sekwencyjnie.

Wewnętrzne bufor mogą przechowywać więcej niż 8 bitów reprezentujących ruch. Po każdym odczytaniu należy sprawdzić bit MOT. Jeśli wciąż występuje to czytamy rejestry aż bit zgaśnie.

- **Delta\_X rejestr 0x03 dostęp: tylko odczyt**

Zmiana położenia w poziomie.

7	6	5	4	3	2	1	0
X <sub>7</sub>	X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>



Odczytanie tego rejestru powoduje jego wyzerowanie.

- **Delta\_X** rejestr **0x04** dostęp: **tylko odczyt**

Zmiana położenia w pionie deltaY.

7	6	5	4	3	2	1	0
Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>



Odczytanie tego rejestru powoduje jego wyzerowanie.

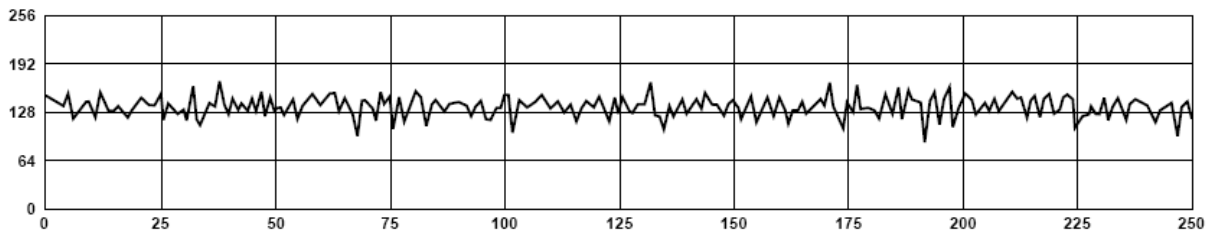
- **SQUAL** rejestr **0x05** dostęp: **tylko odczyt**

Jakość powierzchni.

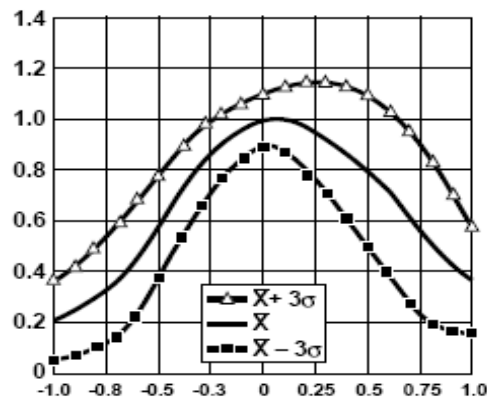
7	6	5	4	3	2	1	0
SQ <sub>7</sub>	SQ <sub>6</sub>	SQ <sub>5</sub>	SQ <sub>4</sub>	SQ <sub>3</sub>	SQ <sub>2</sub>	SQ <sub>1</sub>	SQ <sub>0</sub>

Rejestr dostarcza informacji o jakości powierzchni (widzialności). Można się nim posłużyć do poprawnego ustawienia ogniskowej w osi Z. Wartości przyjmowane są pomiędzy 0 a 255.

Przykład: 250 próbek pobranych podczas ruchu po białej kartce.



Wykres przedstawia wielokrotne pomiary na różnych wysokościach.



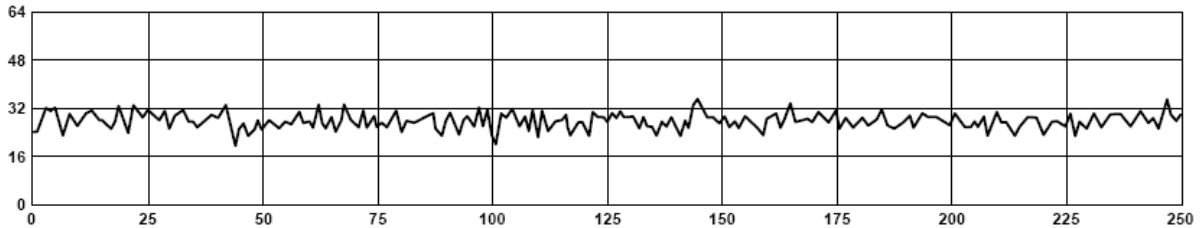
- **Average\_Pixel** rejestr **0x06** dostęp: **tylko odczyt**

Średnia wartość wszystkich pikseli.

7	6	5	4	3	2	1	0
0	0	AP <sub>5</sub>	AP <sub>4</sub>	AP <sub>3</sub>	AP <sub>2</sub>	AP <sub>1</sub>	AP <sub>0</sub>

Rejestr informuje o średniej wartości wszystkich pikseli w danej ramce. Wartości przyjmowane są pomiędzy 0 a 63.

Przykład 250 próbek pobranych podczas ruchu po białej kartce



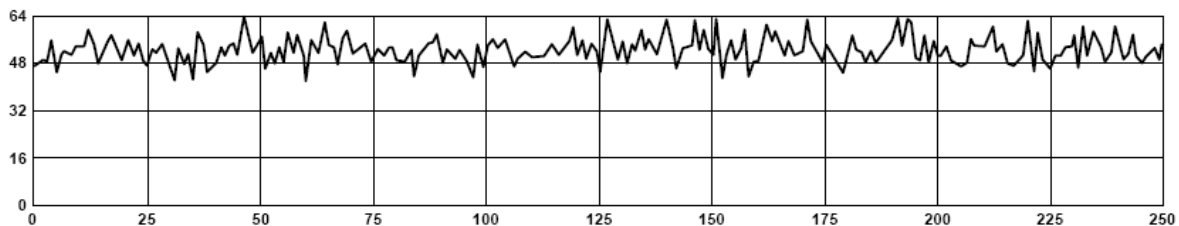
- **Maximum\_Pixel** rejestr **0x07** dostęp: **tylko odczyt**

Maksymalna wartość piksela w obecnej ramce.

7	6	5	4	3	2	1	0
0	0	MP <sub>5</sub>	MP <sub>4</sub>	MP <sub>3</sub>	MP <sub>2</sub>	MP <sub>1</sub>	MP <sub>0</sub>

Rejestr zawiera maksymalną wartość piksela w danej ramce. Wartości przyjmowane są pomiędzy 0 a 63.

Przykład 250 próbek pobranych podczas ruchu po białej kartce



- **Configuration\_bits** rejestr **0x0a** dostęp: **odczyt/zapis**

Bity konfiguracyjne.

7	6	5	4	3	2	1	0
RESET	LED_MODE	Sys Test	RES	PixDump	Reserved	Reserved	Sleep

## RESET

0 – bez akcji.

1- resetuje rejestry i wczytuje domyślne ustawienia.

## LED\_MODE

0 – tryb czas migawki wyłączony, dioda załączona nawet jeśli nie wystąpił ruch przez 1s.

1 – tryb czas migawki załączony, dioda uruchamiana tylko wtedy kiedy otwarto przysłonę.

## SYS\_TEST

0 – bez akcji.

1- testuje cały układ zapisując do pamięci domyślne wartości. Wynik zapisuje do rejestrów Data\_Out\_Upper i Data\_Out\_Lower.

## RES

0 – rozdzielczość 400dpi

1- 800dpi

## PIXDUMP

0 – bez akcji

1- wyślij wartości wszystkich pikseli poprzez rejestry Data\_Out\_Upper i Data\_Out\_Lower.

## SLEEP

0 – przejście w stan uśpienia po 1 sekundzie lub po braku ruchy przez min 1500 ramek.

1- zawsze aktywny.

- **Data\_Out\_Lower** rejestr **0x0c** dostęp: **tylko odczyt**

7	6	5	4	3	2	1	0
DO <sub>7</sub>	DO <sub>6</sub>	DO <sub>5</sub>	DO <sub>4</sub>	DO <sub>3</sub>	DO <sub>2</sub>	DO <sub>1</sub>	DO <sub>0</sub>

- **Data\_Out\_Upper** rejestr **0x0d** dostęp: **tylko odczyt**

7	6	5	4	3	2	1	0
DO <sub>15</sub>	DO <sub>14</sub>	DO <sub>13</sub>	DO <sub>12</sub>	DO <sub>11</sub>	DO <sub>10</sub>	DO <sub>9</sub>	DO <sub>8</sub>

Szesnasto bitowe słowo (0x0c i 0x0d) może być użyte do odczytywania wartości testu lub po wywołaniu poprzez rejestr konfiguracyjny opcji PixelDump. Po wywołaniu opcji PixelDump można kolejno odczytywać wartości wszystkich 256 pikseli. Przy czym w rejestrze Upper będzie zawierać się adres piksela a w Lower jego wartość (tylko 6 dolnych bitów). Kiedy już nastąpi odczytanie wszystkich pikseli bit PixelDump w rejestrze konfiguracyjnym 0x0a zostanie zgaszony. Bit MSB rejestru Lower to bit statusowy mówi o tym czy dane w nim zawarte są prawidłowe. Jeśli bit ustawiony na 1 to znaczy, że dane nie są gotowe. Przy użyciu trybu PixelDump powinno się używać także opcji SLEEP ustawionej na 1 (zawsze aktywny).

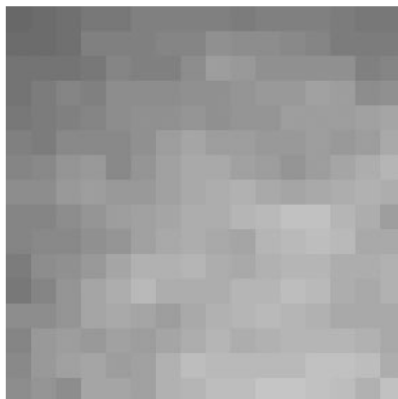
Adresy poszczególnych pikseli:

LAST PIXEL

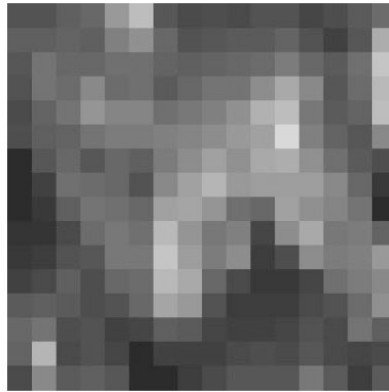
FF	EF	DF	CF	BF	AF	9F	8F	7F	6F	5F	4F	3F	2F	1F	0F
FE	EE	DE	CE	BE	AE	9E	8E	7E	6E	5E	4E	3E	2E	1E	0E
FD	ED	DD	CD	BD	AD	9D	8D	7D	6D	5D	4D	3D	2D	1D	0D
FC	EC	DC	CC	BC	AC	9C	8C	7C	6C	5C	4C	3C	2C	1C	0C
FB	EB	DB	CB	BB	AB	9B	8B	7B	6B	5B	4B	3B	2B	1B	0B
FA	EA	DA	CA	BA	AA	9A	8A	7A	6A	5A	4A	3A	2A	1A	0A
F9	E9	D9	C9	B9	A9	99	89	79	69	59	49	39	29	19	09
F8	E8	D8	C8	B8	A8	98	88	78	68	58	48	38	28	18	08
F7	E7	D7	C7	B7	A7	97	87	77	67	57	47	37	27	17	07
F6	E6	D6	C6	B6	A6	96	86	76	66	56	46	36	26	16	06
F5	E5	D5	C5	B5	A5	95	85	75	65	55	45	35	25	15	05
F4	E4	D4	C4	B4	A4	94	84	74	64	54	44	34	24	14	04
F3	E3	D3	C3	B3	A3	93	83	73	63	53	43	33	23	13	03
F2	E2	D2	C2	B2	A2	92	82	72	62	52	42	32	22	12	02
F1	E1	D1	C1	B1	A1	91	81	71	61	51	41	31	21	11	01
F0	E0	D0	C0	B0	A0	90	80	70	60	50	40	30	20	10	00

FIRST PIXEL

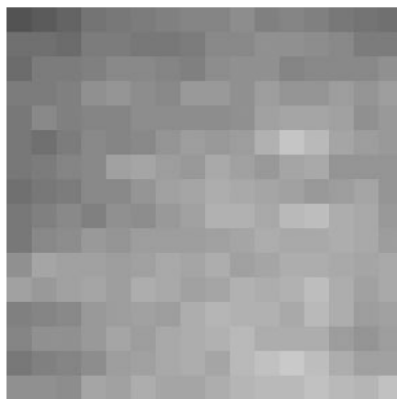
Przykład - biała kartka



Przykład – podkładka pod myszkę



Przykład – szara teczka



- **Shutter\_Lower** rejestr **0x0e** dostęp: **tylko odczyt**

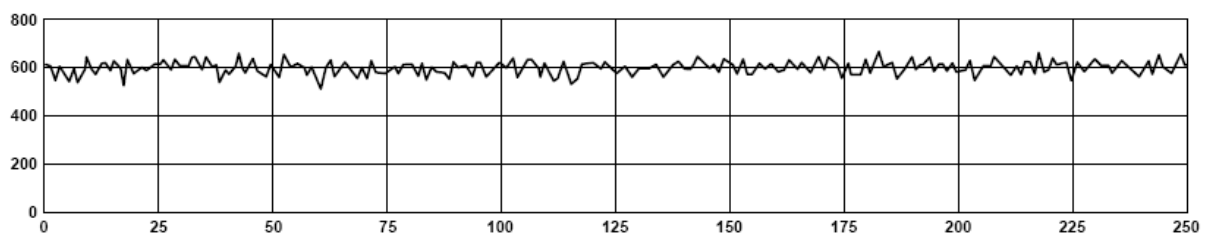
7	6	5	4	3	2	1	0
S <sub>7</sub>	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>

- **Shutter\_Upper** rejestr **0x0f** dostęp: **tylko odczyt**

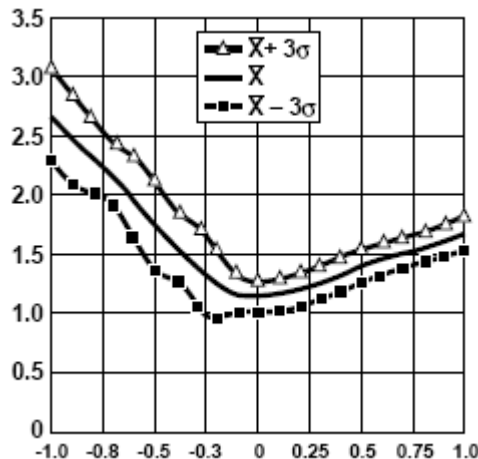
7	6	5	4	3	2	1	0
S <sub>15</sub>	S <sub>14</sub>	S <sub>13</sub>	S <sub>12</sub>	S <sub>11</sub>	S <sub>10</sub>	S <sub>9</sub>	S <sub>8</sub>

Szesnasto bitowe słowo zawarte w dwóch rejestrach 0x0e oraz 0x0f przedstawia wartość czasu migawki i należy czytać sekwencyjnie jeden po drugim. Czas migawki dostraja się na podstawie maksymalnego piksla. Przyjmuje wartość taką aby mieścił się on w całej skali szarości.

Przykład 250 próbek pobranych przy ruchu czujnika po białym papierze.



Wartość czasu migawki zależy również od odległości od powierzchni co przedstawia wykres u dołu.



Maksymalna wartość czasu migawki zależy od zegara systemowego oraz częstotliwości odświeżania.

Wzór jest następujący:

$$\text{Max. Shutter Value} = \frac{\text{Clock Frequency}}{\text{Frame Rate}} - 2816$$

Dla zegara systemowego 18MHz poniższa tabela ukazuje maksymalne wartości migawki.

Frames/second	Max Shutter		Shutter	
	Decimal	Hex	Upper	Lower
2300	5010	0x1392	13	92
2000	6184	0x1828	18	28
1500	9184	0x23E0	23	E0
1000	15184	0x3B50	3B	50
500	33184	0x81A0	81	A0

← Default Max. Shutter

- **Frame\_Period\_Lower** rejestr **0x10** dostęp: **odczyt/zapis**

7	6	5	4	3	2	1	0
FP <sub>7</sub>	FP <sub>6</sub>	FP <sub>5</sub>	FP <sub>4</sub>	FP <sub>3</sub>	FP <sub>2</sub>	FP <sub>1</sub>	FP <sub>0</sub>

- **Frame\_Period\_Upper** rejestr **0x11** dostęp: **odczyt/zapis**

7	6	5	4	3	2	1	0
FP <sub>15</sub>	FP <sub>14</sub>	FP <sub>13</sub>	FP <sub>12</sub>	FP <sub>11</sub>	FP <sub>10</sub>	FP <sub>9</sub>	FP <sub>8</sub>

Szesnasto bitowe słowo zawarte którym definiuje się częstotliwość odświeżania.

Wzór na wyliczenie wartości rejestru jest następujący:

$$\frac{\text{Clock Rate}}{\text{Frame Rate}} = \text{Counts (decimal)} \rightarrow \text{Counts (hex)} \rightarrow \text{Counts (2's complement hex)}$$

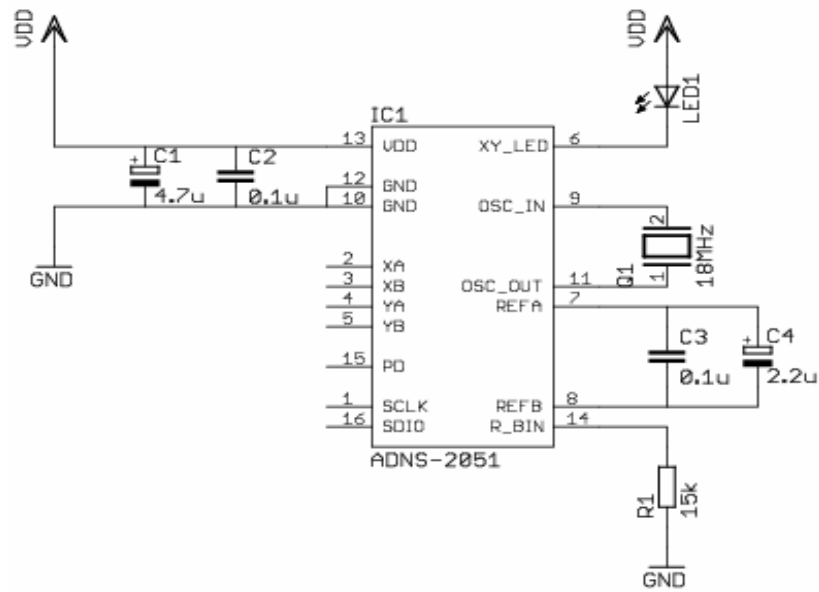


Można również skorzystać z tabelki poniżej przedstawiającej kilka popularnych wartości dla zegara systemowego 18MHz.

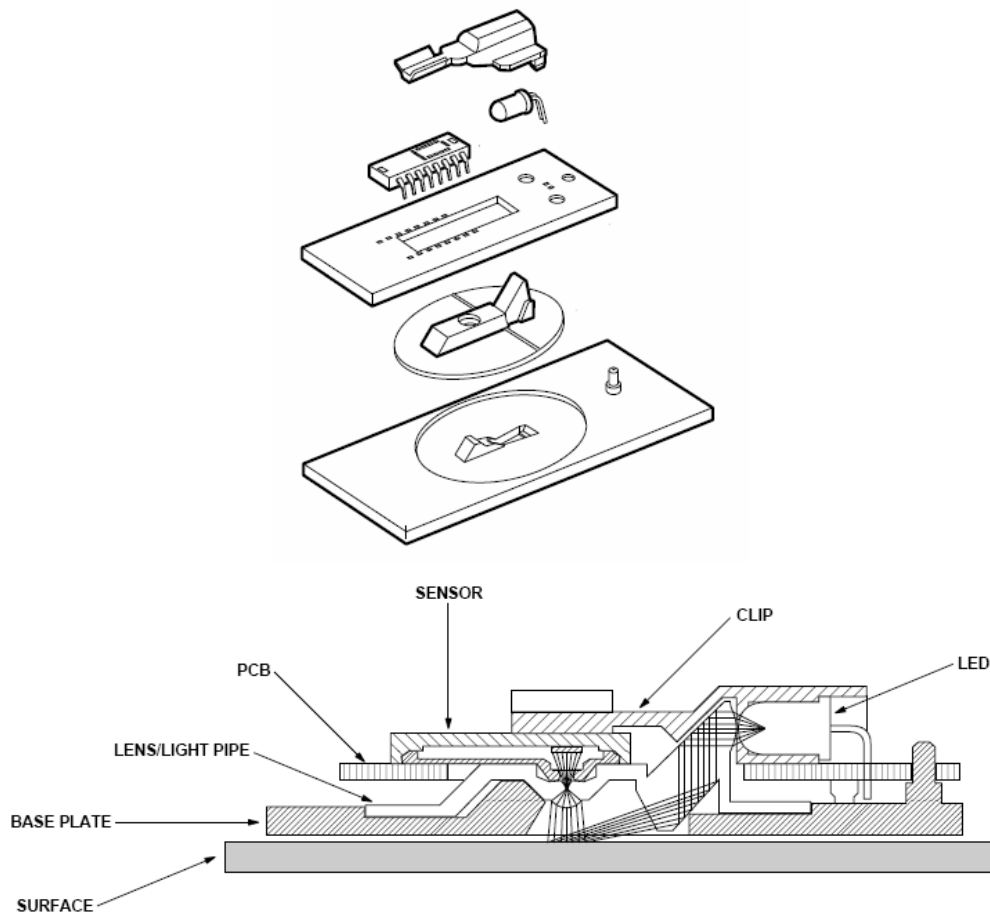
<b>Frames/second</b>	<b>Counts</b>			<b>Frame_Period</b>		
	<b>Decimal</b>	<b>Hex</b>	<b>2's Comp</b>	<b>Upper</b>	<b>Lower</b>	
2300*	7826	0x1E92	0xE16E	E1	6E	
2000*	9000	0x2328	0xDCD8	DC	D8	
1500	12000	0x2EE0	0xD120	D1	20	← Default Frame Time
1000	18000	0x4650	0xB9B0	B9	B0	
500	36000	0x8CA0	0x7360	73	60	← Minimum Frame Time

Gdy dokonujemy odczytu należy najpierw odczytać rejestr Upper następnie Lower. Gdy dokonujemy zapisu czynność tę wykonujemy w odwrotnej kolejności.

## 5 Schemat montażu



Rysunek nr 10 – Schemat montażowy układu ADNS-2051



Rysunek nr 11 – Układ optyczny dla ADNS-2051

## 6 Oprogramowanie

Poniżej przedstawiono sposób realizacji programu napisanego dla mikrokontrolera formy FreeScale MC68332.

### Plik nagłówkowy **mouse.h**

```
void delay(unsigned int); // funkcja delay
unsigned char czytaj_bajt_mouse(); // odczyt bajtu z myszki
void wyslij_bajt_mouse(unsigned char); // zapis bajtu z myszki
void Mouse_Init(); // inicjacja myszki

#define SCLK 4 // definiujemy PIN zegara
#define SDIO 5 // definiujemy PIN wej/wyj

#define CONF_ADR 0x0a // rejestr konfiguracyjny
#define DEV_ID_ADR 0x00 // rejestr ProducID
#define MOTION_ADR 0x02 // rejestr Motion
#define DELTA_X_ADR 0x03 // rejestr Delta_X
#define DELTA_Y_ADR 0x04 // rejestr DeltaY
#define QUALITY_ADR 0x05 // rejestr SQUAL

#define WRITE_MOUSE 0x80 // bit zapisu
#define RESET_MOUSE 0x80 // bit resetu
#define LED_MODE 0x40 // bit ustawienia trybu migawki
#define RES_800 0x10 // bit ustawienia rozdzielczosci

#define wait 3

#define long_wait 100
```

### Plik źródłowy **mouse.c**

```
#include "mouse.h"

void delay(unsigned int c) { while(c-->0); return;}
//delay sie przydaje do czekania miedzy bajtami

/* wyslij osiem bitow danych zaczynajac od najstarszego bitu D7..D0 */
void wyslij_bajt_mouse(unsigned char bajt)
{
    unsigned char maska=0x80; // zacznij od najstarszego bitu
    DDRE|= (1<<SDIO); //PIN SDIO jako wyjście
    do {
        ClrBite(SCLK);
        if (bajt&maska) SetBite(SDIO); else ClrBite(SDIO); // ustawia stan
        delay(wait);
        SetBite(SCLK); // clock 1
        delay(wait);
    } while(maska>>=1); // kolejny bit
    SetBite(SDIO); // przy wyjściu z funkcji ustawia PIN na 1
}
```

```

/* czytaj osiem bitow danych zaczynajac od najstarszego bitu D7..D0 */
unsigned char czytaj_bajt_mouse()
{
    unsigned char maska=0x80,bajt=0x00; /* zacznij od MSB bitu */
    DDRE &= ~(1<<SDIO);                //PIN SDIO jako wejscie
    delay(3);
    do {
        ClrBitE(SCLK);
        delay(wait);
        SetBitE(SCLK);                    // clock 1
        delay(wait);
        if (CzytBitE(SDIO)==1) bajt=bajt|maska; // sprawdza stan PIN-u SDIO
        } while(maska>>=1);             // ustawia kolejny bit
    return bajt;                          // zwraca zczyany bajt
}

void Mouse_Init()
{
    DDRE|= (1<<SCLK);                    //PIN SCLK jako wyjscie

    wyslij_bajt_mouse(WRITE_MOUSE|CONF_ADR); // zapis konfiguracji
    delay(long_wait);                     // zaczekaj
    wyslij_bajt_mouse(RESET_MOUSE);        // Resetuj urzadzenie
    delay(long_wait);                     // zaczekaj

    wyslij_bajt_mouse(WRITE_MOUSE|CONF_ADR); // zapis konfiguracji
    delay(long_wait);                     // zaczekaj
    wyslij_bajt_mouse(RES_800|LED_MODE);    //shuter_on oraz 800dpi
    delay(long_wait);                     // zaczekaj

    wyslij_bajt_mouse(DEV_ID_ADR);         // rejestr ID
    delay(long_wait);                     // zaczekaj
    jakis char=(unsigned char)czytaj_bajt_mouse(); //typ urzadzenia
    delay(long_wait);                     // zaczekaj
}

```

Następnie odczytów dokonujemy np. w przerwaniu lub w programie głównym **main.c**

```

#include "mouse.h"

main()
{
    struct typ_stan          // definiujemy strukturke stanu
    {
        char hold;
        signed long deltaX;
        signed long deltaY;
        unsigned char surface;
        unsigned char motion;
        unsigned char device;
    };
    struct typ_stan stan;
}

```

```

Mouse_Init();           // inicjujemy nasza myszke
while(1){
    // klejno czytamy parametry
    delay(100);
    wyslij_bajt_mouse(QUALITY_ADR);           // wyslij adres
    delay(100);
    stan.surface=(unsigned char)czytaj_bajt_mouse(); // czytaj dane
    delay(100);

    wyslij_bajt_mouse(MOTION_ADR);           // wyslij adres
    delay(100);
    stan.motion=(unsigned char)czytaj_bajt_mouse(); // czytaj dane
    delay(100);

    wyslij_bajt_mouse(DELTA_X_ADR);           // wyslij adres
    delay(100);
    stan.deltaX=(unsigned char)czytaj_bajt_mouse(); // czytaj dane
    delay(100);

    wyslij_bajt_mouse(DELTA_Y_ADR);           // wyslij adres
    delay(100);
    stan.deltaY=(unsigned char)czytaj_bajt_mouse(); // czytaj dane
    delay(100);

    // dane gotowe mozna zatem zapisywac np na LCD
    sprintf(buf,"x:  %6d ",stan. deltaX);
    Lcd_Put_Text(buf);
    sprintf(buf,"y:  %6d ",stan. deltaY);
    Lcd_Put_Text(buf);

    if (stan.motion&0x01)
        Lcd_Put_Text("res.:  800dpi");
    else
        Lcd_Put_Text("res.:  400dpi");

    if (stan.motion&0x80)
        Lcd_Put_Text("move:   yes");
    else
        Lcd_Put_Text("move:   no");

    sprintf(buf,"quality:  %3d", (int)stan.surface);
    Lcd_Put_Text(buf);

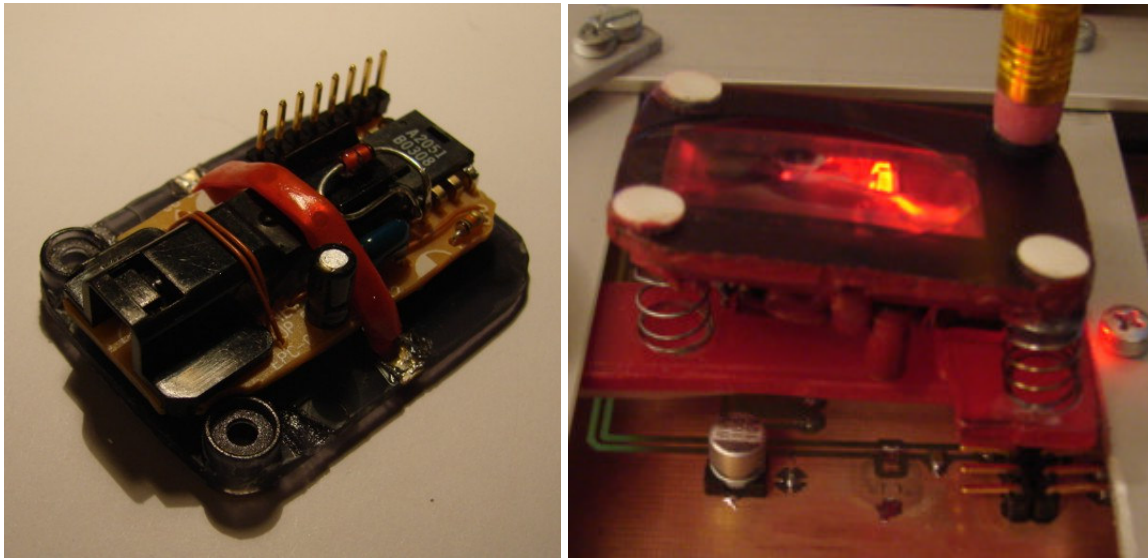
}
}

```

## 6 Realizacja projektu

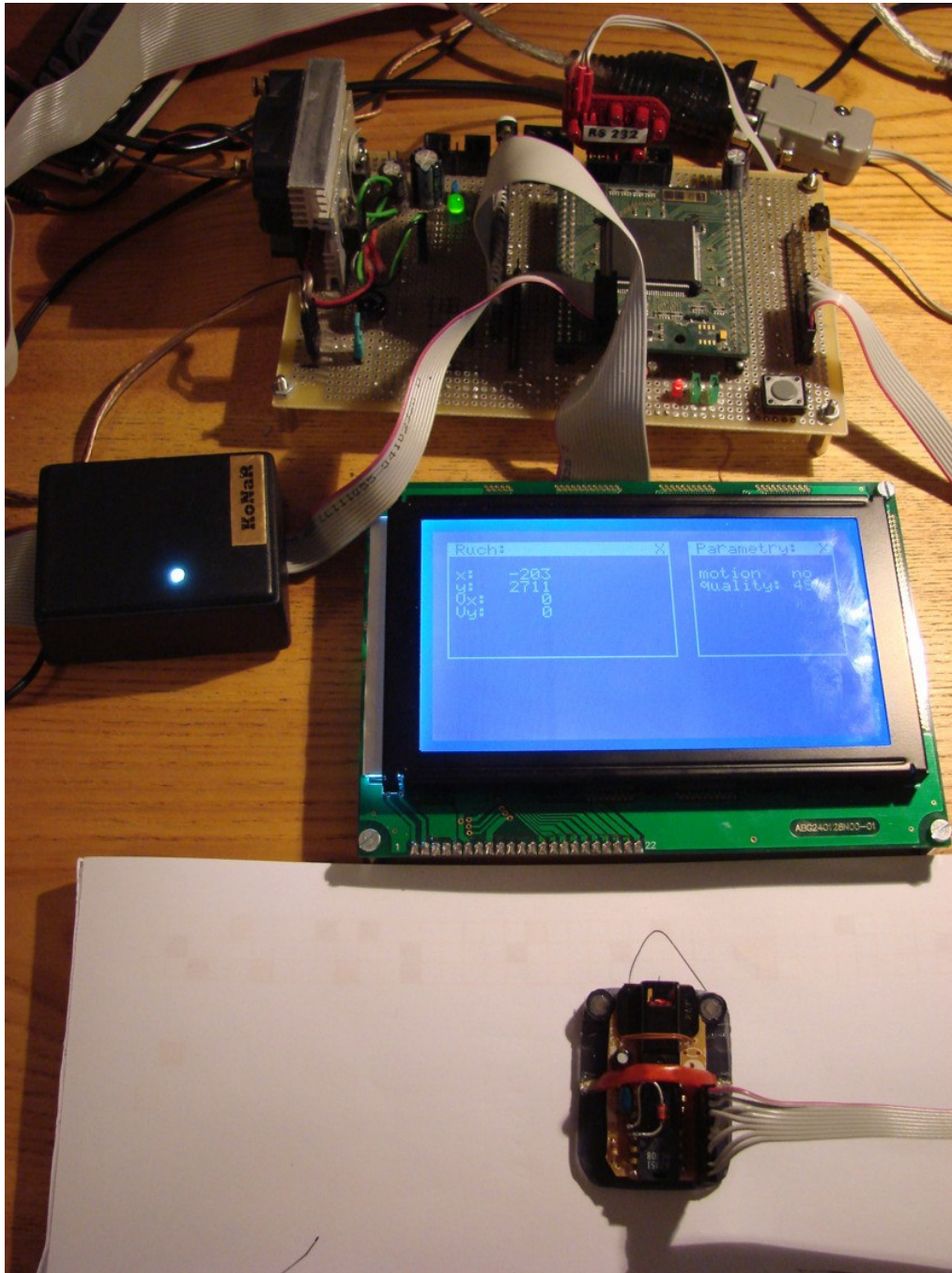
Do komunikacji z ADNS-2051 wykorzystano mikrokontroler MC68332 firmy Freescale (Motorola) [4]. Jednostka ta posiada osobny procesor czasowy tzn. TPU (Time Processor Unit), który umożliwi dokładne badanie przebiegów czasowych na wyjściach kwadraturowych.

Wykorzystując oryginalną płytkę montażową oraz układ optyczny (rys. 11), który znajduje się wewnątrz myszki skonstruowano moduł do testowania czujnika (rys. 12). Takie rozwiązanie zapewnia utrzymanie właściwej odległości czujnika od nawierzchni.



Rysunek nr 12 – Gotowy moduł

Skutecznie udało się zaimplementować wcześniej przedstawioną metodę komunikacji z układem. Przebadano również wyjścia kwadraturowe. Wyniki badań obserwowano na wyświetlaczu LCD (Rys.13) i ( Rys.14) oraz na komputerze przy pomocy interfejsu RS232 (Rys 15).



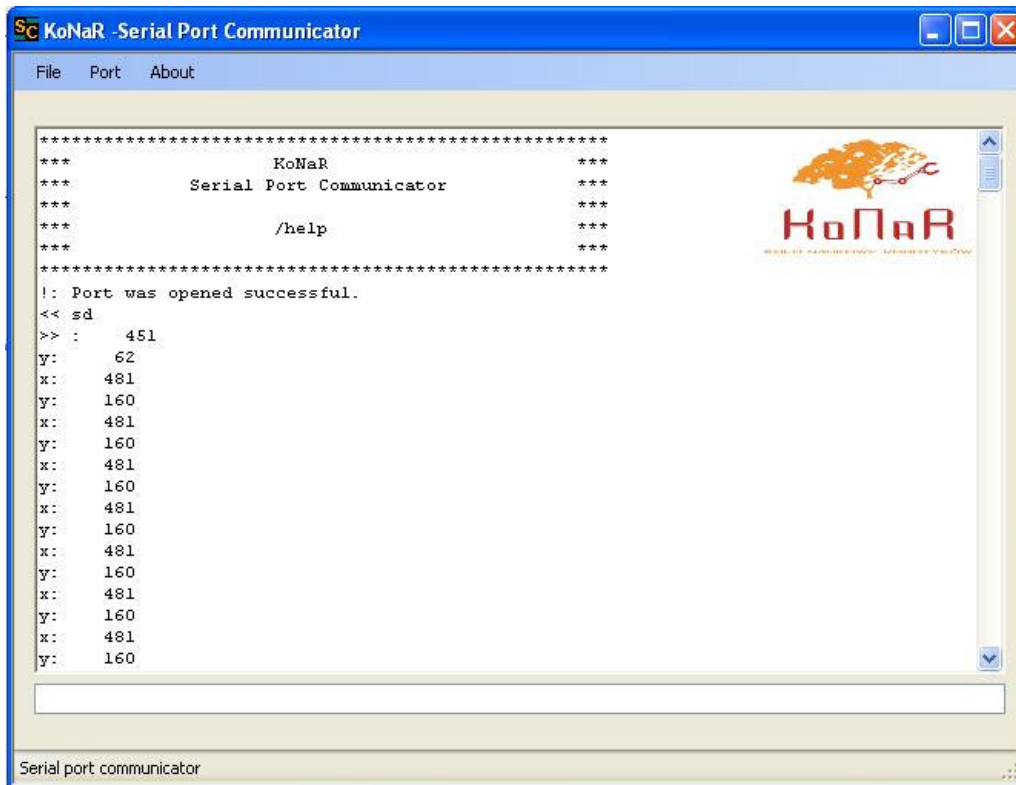
Rysunek nr 13 – Stanowisko pomiarowe.

Osiągnięte rezultaty oraz skastrowany sprzęt umożliwiają dalszą realizację projektu. Wstępne wyniki otrzymane w badaniach testowych wykazują, że wybrany czujnik całkowicie spełnia wymagania projektowe.





Rysunek nr 14 – Dane na LCD.



Rysunek nr 15 – Dane przesłane do komputera.



## 7 Literatura

- [1] *Agilent ADNS-2051 Optical Mouse Sensor, Data Sheet*, Agilent Technologies, 2002
- [2] *Agilent ADNS-2610 Optical Mouse Sensor, Data Sheet*, Agilent Technologies, 2004
- [3] Richard L.Owens, *Optical Mouse technology*  
[www.mstarmetro.net/~rlovens/OpticalMouse/](http://www.mstarmetro.net/~rlovens/OpticalMouse/)
- [4] Wnuk M. *Moduł z Mikrokontrolerem MC68332*, Raport ICT serii 07/2004, Wrocław, 2004