THE GREAT CHEESE HUNT

A NEW MICROMOUSE EVENT

TECHNICAL BULLETIN #5

PUBLICATION RIGHTS RESERVED

ADDRESS QUESTIONS TO:

DR. ANDREW R. SASS

IEEE, L.I. SECTION

arshacker@aol.com

IT IS ASSUMED THAT THE READER IS FAMILIAR WITH MICROMOUSE:THE GREAT CHEESE HUNT - TECH BULLETIN #1, #2, #3, #4.

The Author is grateful to Sean Watts, coach of the Stevens Institute of Technology Micromouse Team, for the key observation that led to this Bulletin.

Tech Bulletin #1, The Great Cheese Hunt, presented a Mouse Sim to be used as the microcomputer hardware and software foundation for a Mouse entry in the newly proposed competition, Micromouse: The Great Cheese Hunt. The Mouse circuit diagram Fig. 1 on Pg. 7 and associated code, MOUSE SIM, shows the use of D2 as a HW Interrupt triggered by the IR Sensor.

As Sean Watts indicated, D2 and D3 are used by many Micromouse Teams for Hardware Interrupts by Wheel Motor encoders. His point is critical in that the Arduino Nano (and Uno) has only two HW Interrupts. Obviously, the use of the HW INTs in the Beacon, as described in Tech Bulletin #1, is not an issue, however for the Mouse, HW Int 2,3 is allocated to Motor Encoders by many teams.

Common knowledge is that the Uno and Nano have only two HW Interrupts (and the Mega has six). This seemingly leads to a difficult if not insurmountable problem for the Cheese Hunt Mouse except for a little-known fact known only to a small number of Arduino users.

The Nano (and Uno) based on Atmega 328p support the use of any pin as a PCINT, an interrupt pin. While a full tutorial of PCINT will not be presented here, a basic tutorial is essential to show its use for our purposes; a THIRD interrupt to be added to HW Interrupts D2,3 to be used for the IR Sensor.

Short Tutorial of PCINT for a Micromouse Nano:

A pin diagram and basic Assembler registers and commands related to PCINT is given in the Appendix at the end of this tutorial.

PCINT is a group interrupt service routine (ISR) activated by a state change of any pin in a Nano Group. Nano has 3 groups; Group 0( Port B), Group 1(Port C), Group 2(Port D).

Any pin in the Nano can be programmed as a PCINT, of course we reserve D2,3 for HWINT.

For this application, once a pin is selected as PCINT, all other pins in the Group/Port must be masked out so they do not trigger the ISR for that Port/Group.

The PCINT\_MASK TEST ino in the repository was used to test this requirement for Pin 10 (Port B), Pin 5 (Port D), and Pin A2(Port D). The code for the A2 test was the last test performed as seen by the relevant #defines. For that test A4 input should not and did not trigger the ISR. This was the required result.

In our example, Pin 10 has been designated as the IR Sensor interrupt pin for MOUSE SIM. Fig. 1 in Tech Bulletin #1 is repeated below, illustrating this change.

A diagram of a circuit

AI-generated content may be incorrect.

and the corresponding Figure 1 from Tech Bulletin #4 showing an RGB LED and current-limiting resistor to reduce component count is also changed to the Figure below.

A diagram of a circuit

AI-generated content may be incorrect.

Mouse Sim RED and GREEN LEDs have been moved from OUTPUT Pins 7,5 as shown in Tech Bulletin # 1, Fig. 1 and Tech Bulletin #4, Fig !, to OUTPUT Pins 9,8 in Port B.

MOUSE\_SIM\_1\_GH sample code in the repository contains these particulars with Pins 12,11 arbitrarily designated as general OUTPUTs, however each can be changed to INPUTS.

In fact, none of the above pin designations is sacred, other than maintaining Pins 2, 3 free for HWINTs. Any other pin may be used for PCINT as long as it is masked to prevent ISR triggering from another pin in its Group. Details for PCINT pin designation and masking is shown in the Appendix below and can be compared to MOUSE\_SIM\_1\_GH sample code. Similarly, Beacon, RED LED and GREEN LED can be assigned to any pins (other than Pins 2,3) which are designated as OUTPUTs.

Besides the Group Interrupt feature of PCINT described above, the other difference with respect to HW INT is that PCINT triggers ISR on a state change of the pin and is not hardware programmed for RISING, FALLING, and CHANGE.

However, PCINT can be software programmed within the ISR for RISING or FALLING.

In this application the ISR will be internally programmed for RISING so that it triggers at the end of the negative pulse of the IR Sensor (as was done in the Tech Bulletin #1, Mouse Sim code which used HWINT).

The changes to Mouse Sim in the repository are as follows.

The activation of the HW ISR in Setup was

attachInterrupt(digitalPinToInterrupt(IR\_SENSOR),IR\_triggerLatch,RISING);//IR SENSOR NEG PULSE

and the HW ISR was (PULSE and LOCK are defined in the code)

void IR\_triggerLatch(){

if(PULSE == LOW){

LOCK = LOW;

}

}

The activation of PIN 10 as the PCINT pin in Setup is now as follows with RISING\_EDGE defined.

bool RISING\_EDGE=1;

(in Setup)

PCICR |= B00000001; //GRP 0

PCMSK0 |= B00000100; //PIN D10 mask

The ISR is now

ISR(PCINT0\_vect){

if((PINB & B00000100) && RISING\_EDGE ){

if(PULSE==LOW){

LOCK = LOW;}

RISING\_EDGE = 0;

}

if(!(PINB & B00000100) && !RISING\_EDGE ){

//LOCK = HIGH; //LOCKED IN LOOP

RISING\_EDGE = 1;

}

}

APPENDIX

NANO PIN/PORT/GROUP DIAGRAM

AND

PCINT ASSEMBLER-REGISTER INSTRUCTIONS

PORT C-GROUP 1

PCINT 8 9 10 11 12 13

ANALOG 0 1 2 3 4 5

NANO

DIG 0 1 2 3 4 5 6 7 8 9 10 11 12 13

PCINT 16 17 18 19 20 21 22 23 0 1 2 3 4 5

PORT D-GROUP 2 PORT B-GROUP 0

PCICR = | B00000DCB

PCMSK0 =| B 0 0 13 12 11 10 9 8

PCMSK2 =| B 7 6 5 4 3 2 1 0

PCMSK1 =| B 0 0 A5 A4A3 A2 A1 A0

PINB, PINC, PIND ARE “PORT INPUT”

EXAMPLE

SETUP

PCICR |= B00000111; //GRP 0,1,2 FROM RIGHT TO LEFT activates PCINT IN ALL GROUPS

BOOOOOOO1 ACTIVATES ONLY GROUP 0

PCMSK0 |= B00000100; //PIN D10 “ “

PCMSK2 |= B00100000; //PIN 5 “ “

PCMSK1 |= B00000100; //pin A2 “ “

ISR(PCINT0\_vect){ //GROUP 0 ISR

if((PINB & B00000100) && RISING\_EDGE ){ //pin 10

LOCK = LOW;

RISING\_EDGE = 0;

}

if(!(PINB & B00000100) && !RISING\_EDGE ){

//LOCK = HIGH; //LOCKED IN LOOP

RISING\_EDGE = 1;

}

}

ISR(PCINT1\_vect){

if((PINC & B00000100) && RISING\_EDGE ){ //pin A2

LOCK1 = LOW;

RISING\_EDGE = 0;

}

if(!(PINC & B00000100) && !RISING\_EDGE ){

//LOCK = HIGH; //LOCKED IN LOOP

RISING\_EDGE = 1;

}

}

ISR(PCINT2\_vect){

if((PIND & B00100000) && RISING\_EDGE ){ Pin 5

LOCK2 = LOW;

RISING\_EDGE = 0;

}

if(!(PIND & B00100000) && !RISING\_EDGE ){

//LOCK = HIGH; //LOCKED IN LOOP

RISING\_EDGE = 1;

}

}