Promoting robotic design and entrepreneurship experiences among students and teachers

Lesson 12: Advanced Arduino Programming - I
• Decimal - binary system
• Math operators

• TASK/ACTIVITY: Do it yourself – example problems
DECIMAL TO BINARY CONVERSION

- **Decimal system:** A numeral system whose numbers are represented with *digits* 0-9
  - The number is expressed as a **base-10** reference system
    - Examples: 2_{10}, 5_{10}, 10_{10}, 100_{10}, etc.

- **Binary system:** A numeral system whose numbers are represented with *digits* 0 and 1 only
  - The number is expressed as a **base-2** reference system
    - Examples: 00_2, 10_2 (10 is *not* ‘number ten’ but equals to ‘number two’ in the base 2 system), 01_2, 010_2 etc.
DECIMAL TO BINARY CONVERSION

- The process of converting a decimal number to binary number is called successive division

- Process:

1. Divide the decimal number by 2
2. The remainder is the **Least Significant Bit (LSB)** of binary number
3. The division is continued until the quotient is zero, then the conversion is complete
4. The new remainder is the next **Most Significant Bit (MSB)** of the binary number in every successive division step
DECIMAL TO BINARY CONVERSION

• Example 1: Convert decimal number 6 (base 10) to binary number (base 2)

\[
\begin{array}{c}
\text{3} \\
\text{2 \big) 6} \\
\text{\underline{2}} \\
\text{\text{Remainder = 0} \rightarrow \text{LSB}} \\
\text{1} \\
\text{2 \big) 3} \\
\text{\underline{2}} \\
\text{\text{Remainder = 1} \rightarrow \text{next MSB}} \\
\text{0} \\
\text{2 \big) 1} \\
\text{\underline{2}} \\
\text{\text{Remainder = 1} \rightarrow \text{MSB}} \\
\end{array}
\]

\[\therefore 6_{10} = 110_2\]
Problem 1: Do it yourself!

Convert the decimal number $26_{10}$ into its binary equivalent
Problem 1: Convert the decimal number $26_{10}$ into its binary equivalent

Solution:

\[
\begin{array}{c|c|c|c}
\text{Quotient} & \text{Remainder} & \text{Bit} \\
\hline
13 & 0 & \text{LSB} \\
6 & 1 & \text{next MSB} \\
3 & 0 & \text{LSB} \\
1 & 1 & \text{MSB} \\
0 & 1 & \\
\end{array}
\]

\[
\therefore \ 26_{10} = 11010_2
\]
Problem 2: Do it yourself!

Convert the decimal number $41_{10}$ into its binary equivalent
Problem 2: Convert the decimal number $41_{10}$ into its binary equivalent

Solution:

\[
\begin{array}{c|c}
20 & 41 \\
\hline
10 & 20 \\
\hline
5 & 10 \\
\hline
2 & 5 \\
\hline
1 & 2 \\
\hline
0 & 1 \\
\end{array}
\]

Remainder = 1 → LSB

Remainder = 0 → next MSB

Remainder = 0

Remainder = 1

Remainder = 0

Remainder = 1 → MSB

$\therefore 41_{10} = 101001_2$
ACTIVITY

a) \( 13_{10} = ? \)
b) \( 22_{10} = ? \)
c) \( 43_{10} = ? \)
d) \( 158_{10} = ? \)
DEC → BINARY: SOLUTIONS

a) $13_{10} = ? \ 1101_2$

b) $22_{10} = ? \ 10110_2$

c) $43_{10} = ? \ 101011_2$

d) $158_{10} = ? \ 10011110_2$
• The process of converting a binary number to decimal number is called **weighted multiplication**

• **Process:**
  1. The decimal number will be equal to the sum of each binary digit starting from the left-most digit times their power of 2 \( (2^n) \), \( n = 0, 1, 2, .., (n – 1) \) which is called the **bit-weighting factor**, where \( n \) is the **no. of digits** in the **binary number**
  2. Example: Bit 0 \( \rightarrow 2^0 = 1 \), Bit 1 \( \rightarrow 2^1 = 2 \), Bit 2 \( \rightarrow 2^2 = 4 \), etc.
Example:
Convert the decimal number $0110_2$ into its decimal equivalent where $n = 4$

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>$2^3 = 8$</td>
<td>$2^2 = 4$</td>
<td>$2^1 = 2$</td>
<td>$2^0 = 1$</td>
<td></td>
</tr>
<tr>
<td>$\parallel$</td>
<td>$\parallel$</td>
<td>$\parallel$</td>
<td>$\parallel$</td>
<td>$\parallel$</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>$6_{10}$</td>
</tr>
</tbody>
</table>
Problem 1: Do it yourself!

Convert the binary number $10010_2$ into its decimal equivalent where $n = 6$
**Problem 1:** Convert the binary number 10010 into its decimal equivalent where \( n = 5 \)

**Solution:**

<table>
<thead>
<tr>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

\[
egin{align*}
2^4 &= 16 \\
2^3 &= 8 \\
2^2 &= 4 \\
2^1 &= 2 \\
2^0 &= 1
\end{align*}
\]

\[
\begin{align*}
\underline{16} & \quad + \quad 0 \quad + \quad 0 \quad + \quad 2 \quad + \quad 0 \quad = \quad 18_{10}
\end{align*}
\]

\[ \therefore 10010_2 = 18_{10} \]
BINARY $\rightarrow$ DEC: ACTIVITY

a) $0110_2 = ?$

b) $11010_2 = ?$

c) $0110101_2 = ?$

d) $11010011_2 = ?$
a) $0110_2 = ? \quad 6_{10}$

b) $11010_2 = ? \quad 26_{10}$

c) $0110101_2 = ? \quad 53_{10}$

d) $11010011_2 = ? \quad 211_{10}$
**DECIMAL AND BINARY – REVIEW**

Base\textsubscript{10} \hspace{1cm} \textbf{DECIMAL} \hspace{1cm} \textbf{Successive Division} \hspace{1cm} \textbf{Base\textsubscript{2}} \hspace{1cm} \textbf{BINARY}

\textbf{a)} \hspace{0.5cm} Divide the decimal number by 2, remainder = LSB of binary number

\textbf{b)} \hspace{0.5cm} If the quotient = zero, the conversion is complete; else repeat step (a) using the quotient = decimal number and new remainder = next MSB of the binary number

Base\textsubscript{2} \hspace{1cm} \textbf{BINARY} \hspace{1cm} \textbf{Weighted Multiplication} \hspace{1cm} \textbf{Base\textsubscript{10}} \hspace{1cm} \textbf{DECIMAL}

\textbf{a)} \hspace{0.5cm} Multiply each bit of the binary number by its corresponding bit-weighting factor (i.e., Bit - 0 \rightarrow 2^0 = 1, Bit - 1 \rightarrow 2^1 = 2, etc.)

\textbf{b)} \hspace{0.5cm} \textbf{Sum} up all the products in step (a) to get the decimal number
MATH OPERATORS

- **abs(x):** absolute value of x (applicable if x = int and long)
- **fabs(x):** absolute value of x (applicable if x = float)
- **min(x, y) and max(x, y):** minimum and maximum of two arguments x and y
- **pow(x, y):** \( x^y \)
- **sq(x):** \( x^2 \)
- **Trigonometric functions: sin(x), cos(x),** x has to be in radians
- **random(max):** generates a random number between 0 and max
- **random(min, max):** generates a random number between min and max
1. Write a program to display minimum of the absolute values of 40 and -30

1. Write a program to find the value of \( \sin(x) + (\cos(x))^2 \) at \( x = 1.5 \) radians
The minimum value of the array is: -40
int angle=1.5; // in radians
int val;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  val = sin(angle) + sq(cos(angle));
  Serial.print("The value of the equation at 1.5 radians is: ");
  Serial.println(val);
}

void loop() {}
PROGRAMMING: OTHERS

- delay()
- delayMicroseconds
- millis()
- micros()

- delay in milliseconds
- delay in microseconds
- Time in milliseconds since the Arduino board began running the current program
- Time in microseconds since the Arduino board began running the current program
const int ledPin = 3; // the LED pin number connected
int ledState = LOW;   // set the LED state initially LOW
unsigned long previousMillis = 0; // will store last time LED was blinked
const long period = 1000;  // period at which to blink in ms

void setup() {
  pinMode(ledPin, OUTPUT); // set ledpin as output
}

void loop() {
  unsigned long currentMillis = millis(); // store the current time
  if (currentMillis - previousMillis >= period) {  // check if 1000ms passed
    previousMillis = currentMillis; // save the last time you blinked the LED
    if (ledState == LOW) { // if the LED is off turn it on and vice-versa
      ledState = HIGH;
    } else {
      ledState = LOW;
    }
    digitalWrite(ledPin, ledState); // set LED with ledState to blink again
  }
}
Thank You!

Questions and Feedback?