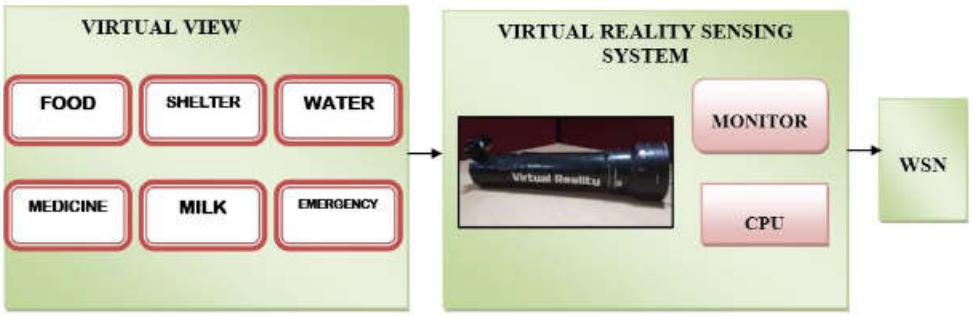
VIRTUAL REALITY BASED SYSTEM FOR FLOOD RISK MANAGEMENT

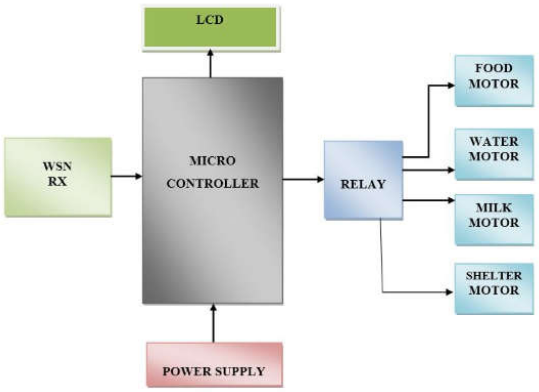
[1]Bairam Ranjith Kumar, [2]Pothuganthi Kiran Kumar, [3]Amgoth Ritesh Naik, [4]Mohammad Sharif, [5]P Shoba, [6]V Ravi Kumar
[1,2,3,4]UG Student, [5]Assistant Professor, [1,2,3,4]Dept. Electrical and Electronics Engineering, Visvesvaraya College of Engineering and Technology, Mangalpalle, Telangana, India

ABSTRACT
Virtual Reality is seen as the high-end of human-computer interactions and it has the potential to target a wide range of applications .During the flood, it is difficult to communicate with the people by the rescue team from the helicopter.It is not possible to interact with everyone directly.Because there won't be any mode of communication during the disaster. The VR system helps to communicate with the people. Redirected walking algorithms require the prediction of human motion in order to effectively steer users away from the boundaries of the physical space. While a virtual trajectory may be represented using straight lines connecting waypoints of interest, this simple model does not accurately represent typical user behavior. We implemented the model within a framework that can be used for redirect walking within different virtual and physical environments. It is useful for the evaluation of redirected of parameters under varying conditions. In the proposed system a projection is made from the helicopter to the ground.When a person stands on a specified location, that particular message is gathered.This helps to communicate with the affected people.

Transmitter Section



Receiver Section



EMBEDDED SYSTEMS
Many embedded systems have substantially different design constraints than desktop computing

applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. Embedded systems in many cases must be optimizedfor life-cycle and business-driven factors rather than for maximum computing throughput. There is currently little *tool* support for expanding embedded computer design to the scope of holistic embedded system design. However, knowing the strengths and weaknesses of current approaches can set expectations appropriately, identify risk areas to tool adopters, and suggest ways in which tool builders can meet industrial needs. If we look around us, today we see numerous appliances which we use daily, be it our refrigerator, the microwave oven, cars, PDAs etc. Most appliances today are powered by something beneath the sheath that makes them do what they do. These are tiny microprocessors, which respond to various keystrokes or inputs. These tiny microprocessors, working on basic assembly languages, are the heart of the appliances. We call them embedded systems. Of all the semiconductor industries, the embedded systems market place is the most conservative, and engineering decisions here usually lean towards established, low risk solutions. Welcome to the world of embedded systems, of computers that will not look like computers and won't function like anything we are familiar with.

CLASSIFICATION

Embedded systems are divided into autonomous, realtime, networked & mobile categories.

Autonomous systems

They function in standalone mode. Many embedded systems used for process control in manufacturing units& automobiles fall under this category.

Real-time embedded systems

These are required to carry out specific tasks in a specified amount of time. These systems are extensively used to carry out time critical tasks in process control.

Networked embedded systems

They monitor plant parameters such as temperature, pressure and humidity and send the data over the network to a centralized system for on line monitoring.

Mobile gadgets

Mobile gadgets need to store databases locally in their memory. These gadgets imbibe powerful computing & communication capabilities to perform realtime as well as nonrealtime tasks and handle multimedia applications. The embedded system is a combination of computer hardware, software, firmware and perhaps additional mechanical parts, designed to perform a specific function. A good example is an automatic washing machine or a microwave oven. Such a system is in direct contrast to a personal computer, which is not designed to do only a specific task. But an embedded system is designed to do a specific task with in a given timeframe, repeatedly, endlessly, with or without human interaction.

Hardware

Good software design in embedded systems stems from a good understanding of the hardware behind it. All embedded systems need a microprocessor, and the kinds of microprocessors used in them are quite varied. A list of some of the common microprocessors families are: ARM family, The Zilog Z8 family, Intel 8051/X86 family, Motorola 68K family and the power PC family. For processing of information and execution of programs, embedded system incorporates microprocessor or micro- controller. In an embedded system the microprocessor is a part of final product and is not available for reprogramming to the end user.

An embedded system also needs memory for two purposes, to store its program and to store its data. Unlike normal desktops in which data and programs are stored at the same place, embedded systems store data and programs in different memories. This is simply because the embedded system does not have a hard drive and the program must be stored in memory even when the power is turned off. This type of memory is called ROM. Embedded applications commonly employ a special type of ROM that can be programmed or reprogrammed with the help of special devices.

Introduction to the Arduino NANOBoard

The Arduino Nano, as the name suggests is a compact, complete and bread-board friendly microcontroller board. The Nano board weighs around 7 grams with dimensions of 4.5 cms to 1.8 cms (L to B). This article discusses about the technical specs most importantly the pinout and functions of each and every pin in the Arduino Nano board.

Arduino Nano has similar functionalities as Arduino Duemilanove but with a different package. The Nano is inbuilt with the ATmega328P microcontroller, same as the Arduino UNO. The main difference between them is that the UNO board is presented in PDIP (Plastic Dual-In-line Package) form with 30 pins and Nano is available in TQFP (plastic quad flat pack) with 32 pins. The extra 2 pins of Arduino Nano serve for the ADC functionalities, while UNO has 6 ADC ports but Nano has 8 ADC ports. The Nano board doesn't have a DC power jack as other Arduino boards, but instead has a mini-USB port. This port is used for both programming and serial monitoring. The fascinating feature in Nano is that it will choose the strongest power source with its potential difference, and the power source selecting jumper is invalid.

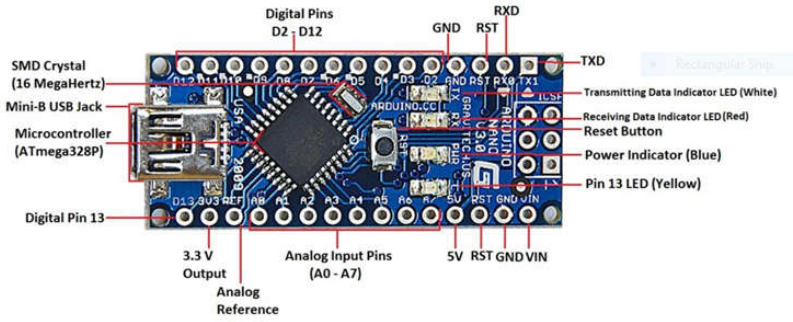


Figure 3.3.2 Arduino nano Board

LCD (Liquid Cristal Display)

Introduction:

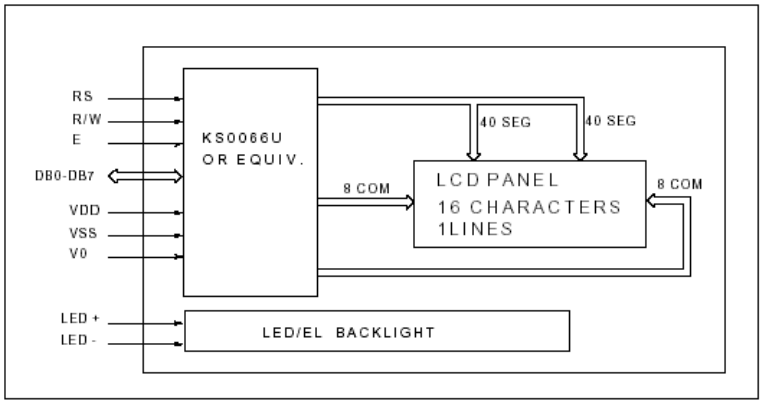
A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a

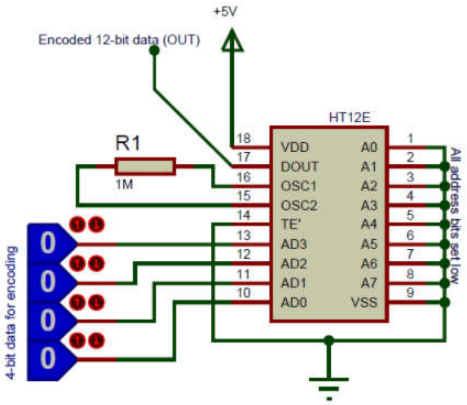
readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

Electrical block diagram:



How to use a HT12E Encoder IC

The primary function of HT12E is to encode a 12-bit and send it out through the output pin. Since the IC comes with an in-built Oscillator it is very easy to make this IC work. The IC is has a wide range of operating voltage from 2.4V to 12V, but normally the Vcc pin (pin 18) is powered by +5V and the ground pin (pin 9) is grounded. Pull the Transmission Enable pin (pin 14) to ground to activate transmission. For decoding a data the IC will requires an oscillator, luckily this IC has one in-built. We just have to connect the OSC1 and OSC2 (pin 15 & 16) through a 1M resistor to invoke it. The 4-bit data that has to be sent has to be given to the pins AD0 to AD1 and an address of 8-bit has to be set using the pins A0 to A7. It is very important that your Decoder should also have this same address for them to talk to each other. A basic **connection diagram for the HT12E IC** is shown below



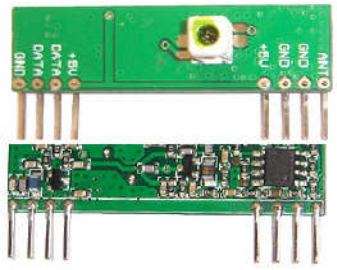
In the above circuit I have set the 8-bit address data as 0b00000000, by connecting all address pins to ground. If you want security you can connect any of the 8 pins to 5V to make it high. The complete IC is powered by a +5V supply which could be obtained from a voltage regulator like 7805.The pins AD3, AD2, AD1 and AD0 are connected to any Digital IC which can provide the 4-bit data. They can also be connected to switches to manually send and receive data. In this image I have made all four data bits as zero (low), when this is decoded we will get the same kind of bits on the output side of HT12D, similarly we can make any changes on these 4-bits and they will be reflected on the output side of the HT12D decoder IC.

The encoded 12-bit can be obtained from the Dout pin (pin 17). This data should be sent to the HT12D for decoding, it can either be sent directly through a wire or by using a wireless medium like RF or IR. You can know to set up the HT12D after this from here.

RF RECEIVER

Overview

The STR-433 is ideal for short-range remote control applications where cost is a primary concern. The receiver module requires no external RF components except for the antenna. It generates virtually no emissions, making FCC and ETSI approvals easy. The super-regenerative design exhibits exceptional sensitivity at a very low cost. The manufacturing-friendly SIP style package and low-cost make the STR-433 suitable for high volume applications.



Features

· Low Cost
· 5V operation
· 3.5mA current drain
· No External Parts are required
· Receiver Frequency: 433.92 MHZ
· Typical sensitivity: -105dBm
· IF Frequency: 1MHz

Applications

· Car security system
· Sensor reporting
· Automation system
· Remote Keyless Entry (RKE)
· Remote Lighting Controls
· On-Site Paging
· Asset Tracking
· Wireless Alarm and Security Systems
· Long Range RFID
· Automated Resource Management

SOFTWARE DESCRIPTION

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs of the UNO board in a smaller form factor.

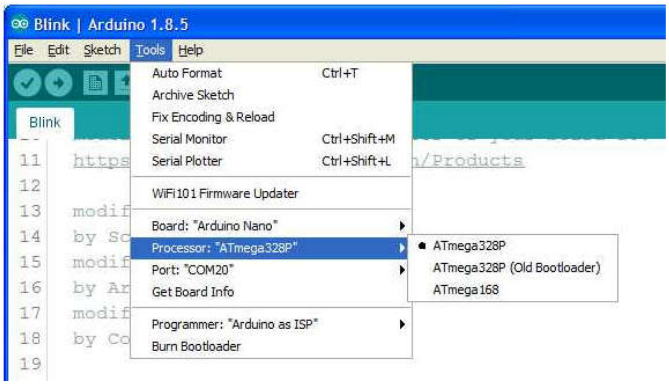
The Arduino Nano is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards

Use your Arduino Nano on the Arduino Desktop IDE

If you want to program your Arduino Nano while offline you need to install the Arduino Desktop IDE To connect the Arduino Nano to your computer, you'll need a Mini-B USB cable. This also provides power to the board, as indicated by the blue LED (which is on the bottom of the Arduino Nano 2.x and the top of the Arduino Nano 3.0).

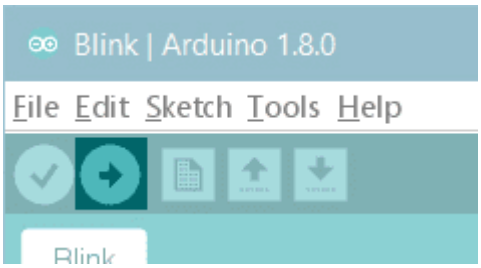
Select your board type and port

Select **Tools > Board > Arduino AVR Boards > Arduino Nano**.



Upload and Run your first Sketch

To upload the sketch to the Arduino Nano, click the **Upload** button in the upper left to load and run the sketch on your board:



Wait a few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar

CONCLUSION

The emergence of virtual reality platform-based technologies applied to disaster preparedness and response training offers significant potential advantages over other traditional forms of training, and is gaining increasing acceptance. • In future virtual reality can play a major role in risk asset management. • one can make a better plan for withstand the flood severity in the scene and go for decision making. • VR is useful in planning of city and damage assessment • Comparative research between VR-based and traditional modalities of disaster training is needed to explore the various aspects of realism, cost, and ultimately disaster readiness

References

1. M. A. Rizaty, "There are 1733 natural disasters in Indonesia to mid 2022", *databoks*, Sep 2022
2. R. Lammers, A. Li, S. Nag and V. Ravindra, "Prediction models for urban flood evolution for satellite remote sensing", *J. Hydrol.*, vol. 603, pp. 127175, Dec. 2021.
3. M. Postacchini, G. Bernardini, M. D'Orazio and E. Quagliarini, "Human stability during floods: Experimental tests on a physical model simulating human body", *Saf. Sci.*, vol. 137, pp. 105153, May 2021.
4. G. Bernardini, M. Postacchini, E. Quagliarini, M. Brocchini, C. Cianca and M. D'Orazio, "A preliminary combined simulation tool for the risk assessment of pedestrians flood-induced evacuation", *Environ. Model. Softw.*, vol. 96, pp. 14-29, Oct. 2017.