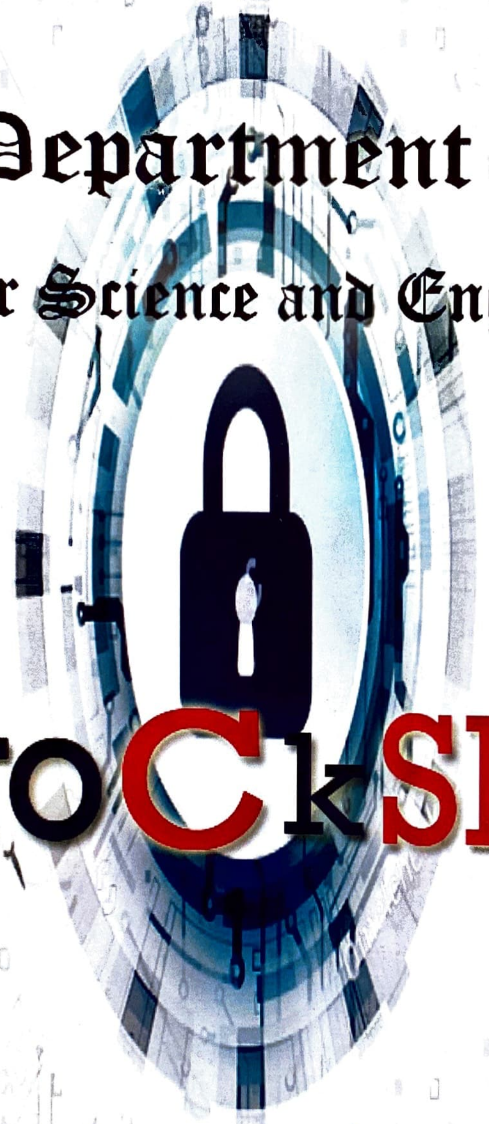
KAKATIYA INSTITUTE OF TECHNOLOGY AND SCIENCE

(An Autonomous Institute Under Kakatiya University)

Warangal-506015, Telangana India

Department of

Computer Science and Engineering



roCkSE

Volume-VIII

March-2018

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE

WARANGAL - 506015

Estd : 1980

Department of Computer Science & Engineering

Presents....

roCkSE

A Technical Magazine

VOLUME-VIII

March 2018

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE
(An Autonomous Institute under Kakatiya University, Warangal)
Opp: Yerragattu Hillock, Vill: Bheemaram, Mdl: Hasanparthy, Warangal: 506015

Department of Computer Science & Engineering

Vision:

Attaining centre of excellence status in various fields of Computer Science and Engineering by offering worth full education, training and research to improve quality of software services for ever growing needs of the industry and society.

Mission:

1. Practice qualitative approach and standards to provide students better understanding and profound knowledge in the fundamentals and concepts of computer science with its allied disciplines.
2. Motivate students in continuous learning to enhance their technical, communicational, and managerial skills to make them competent and cope with the latest trends, technologies, and improvements in computer science to have a successful career with professional ethics.
3. Involve students in analyze, design and experimenting with contemporary research problems in computer science to impact socio-economic, political and environmental aspects of the globe.

Program Educational Objectives (PEOs) - B.Tech

I. Graduates with fundamental knowledge should escalate the technical skills within and across disciplines of Computer Science Engineering for productive career by maintaining professional ethics.

II. Graduates should develop and exercise their capabilities to demonstrate their creativity in engineering practice and exhibit leadership with responsibility in teamwork.

III. Graduates should refine their knowledge and skills to attain professional competence through life-long learning such as higher education, research and professional activities.

Program Specific Outcomes (PSOs) - B.Tech

I. **Software Development and Quality assurance :** Transform various legacy or manual systems into computer automated systems using Modern Programming Languages, Integrated Development Environments, and apply Testing Tools for efficient verification and validation of those software systems.

II. **Maintenance :** Demonstrate knowledge in fixing and updating multidisciplinary software problems working in real time environment.

III. Work as a software practitioner or continue higher education by adopting advanced technologies in various fields of computer science and Engineering.

Program Outcomes (POs)- B.Tech :

I. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

II. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

III. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

IV. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

V. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

VI. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

VII. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

VIII. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

IX. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

X. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

XI. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

XII. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE

(An Autonomous Institute under Kakatiya University, Warangal)

Department of Computer Science & Engineering

Program Outcomes: M.Tech

I. **Engineering Knowledge :** An ability to independently carry out research/ investigation and development Work to solve practical problems.

II. **Problem Analysis:** An ability to write and present a substantial technical report / document

III. **Design/Development of solutions:** Students should be able to demonstrate a degree of mastery over as per the Specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program.

Program Specific Outcomes: M.Tech

I. **Software Development and Quality Assurance :** Apply the Knowledge and current technologies of software engineering to Pursue Research over complex problems of Computer Science domains.

II. **Maintenance :** Equipped with the Industry Ready, Teaching Skills and Entrepreneurship Capabilities.

III. **Immediate Professional Practice:** plan, manage and assess effectively the software products by using the Software Engineering Concepts and Methodologies.

Department of Computer Science & Engineering

Editorial Board

Chief Editor Message

This magazine summarizes the current state of Computer Science & Engineering, latest technologies and also information of department. Providing an arena for the student community to showcase their technical talents is a great task. We took up the challenge to bring awareness to everyone in laying their career steps towards latest technologies. Keeping in view of the present era of technological revolution in the field of Engineering, the CSE department of KITSW presents you **roCkSE.**

Dr. P. Niranjan Reddy

Director Message

I congratulate the team of the faculty members and the students for their brilliant efforts. I wish all the students and faculty a great career ahead. The main focus of the institution is to empower students with sound knowledge, wisdom, experience and training both at the academic level of Engineering and in the highly competitive global industrial market.

We wish the best for all our students, and the members of the institution who reiterate their aims at providing the best in academic and extra-curricular fields.

Dr. Y. Manohar

Principal Message

I am grateful with the idea of this newsletter .It gives overall views on the current evolving technologies which are trending in computer science field.

Dr. P. Venkateshwara Rao

Chief Editor:

Dr. P. Niranjan Reddy(Head Dept. CSE)

Editor In-Charge:

S. Nagaraju(Associate Professor)

Faculty In-Charge:

M. Preethi (Asst. Professor)

S. Madhavi Sudha (Asst. Professor)

Student Editorial Board:

G. Sathwik Reddy (B15CS135)

K. Ranjith (B15CS042)

G. Manikanta (B15CS063)

M. Sai Karthik (B15CS065)

Smart Gardening System

Internet of things is a widely used concept for implementing smart connected cities and homes. Smart cities and homes make living more efficient by eliminating human to machine interaction. It finds its applications in a wide range of markets. One of the applications is Smart Gardening System. Many homeowners and commercial buildings have large gardens, which require maintenance, such as watering. This maintenance can become very cumbersome and expensive for businesses (or homeowners) where these plants are more of 4 periphery than 4 business focus. In the world of advance electronics, life of human beings should be simpler. Hence to make life simpler and convenient, we have made "SMART GARDENING SYSTEM". The aim of this project is to reduce the overhead of maintenance of these gardens through automation of the watering process. It will help save time, money and help the environment through reducing water loss due to overwatering and inefficient spraying. In this smart gardening System, Soil Moisture Sensor checks the moisture level in the soil and sends values to the cloud every 15 seconds. User can switch OFF/ ON the motor through the android application. If moisture level is high, then NodeMCU switches OFF a relay to cut water flow to the plant. Water pump gets automatically off when system finds enough moisture in the soil. Whenever system switched ON or OFF the pump, a status is updated to the user via an android application that is connected to the cloud module, updating the status of water pump and soil moisture. This system is very useful in farms, gardens, home etc. Current system is semi-automated and there is need for little human intervention. In next phase, two modes of operations are expected to be provided, that will help user to work in "no human intervention" (Automated) and "semi user intervention" (Controlled) modes.

-Bharagava Patel

3D optical data storage

It is any form of optical data storage in which information can be recorded or read with three-dimensional resolution (as opposed to the two-dimensional resolution afforded, for example, by CD).

This innovation has the potential to provide petabyte-level mass storage on DVD-sized discs (120 mm). Data recording and read back are achieved by focusing lasers within the medium. However, because of the volumetric nature of the data structure, the laser light must travel through other data points before it reaches the point where reading or recording is desired. Therefore, some kind of nonlinearity is required to ensure that these other data points do not interfere with the addressing of the desired point.

No commercial product based on 3D optical data storage has yet arrived on the mass market, although several companies are actively developing the technology and claim that it may become available 'soon'.

Current optical data storage media, such as the CD and DVD store data as a series of reflective marks on an internal surface of a disc. In order to increase storage capacity, it is possible for discs to hold two or even more of these data layers, but their number is severely limited since the addressing laser interacts with every layer that it passes through on the way to and from the addressed layer. These interactions cause noise that limits the technology to approximately 10 layers. 3D optical data storage methods circumvent this issue by using addressing methods where only the specifically addressed voxel (volumetric pixel) interacts substantially with the addressing light. This necessarily involves nonlinear data reading and writing methods, in particular nonlinear optics.

3D optical data storage is related to (and competes with) holographic data storage. Traditional examples of holographic storage do not address in the third dimension, and are therefore not strictly "3D", but more recently 3D holographic storage has been realized by the use of micro holograms. Layer-selection multilayer technology (where a multilayer disc has layers that can be individually activated e.g. electrically) is also closely related.

-Maniteja (B15CS063)

Swarm robotics

It is an approach to the coordination of multirobot systems, which consist of large numbers of mostly simple physical robots. It is supposed that a desired collective behavior emerges from the interactions between the robots and interactions of robots with the environment. This approach emerged on the field of artificial swarm intelligence, as well as the biological studies of insects, ants and other fields in nature, where swarm behavior occurs.

The research of swarm robotics is to study the design of robots, their physical body and their controlling behaviours. It is inspired but not limited by the emergent behaviour observed in social insects, called swarm intelligence. Relatively simple individual rules can produce a large set of complex swarm behaviours. A key-component is the communication between the members of the group that build a system of constant feedback. The swarm behaviour involves constant change of individuals in cooperation with others, as well as the behaviour of the whole group.

Unlike distributed robotic systems in general, swarm robotics emphasizes a large number of robots, and promotes scalability, for instance by using only local communication. That local communication for example can be achieved by wireless transmission systems, like radio frequency.

Potential applications for swarm robotics are many. They include tasks that demand miniaturization (nano robotics, microbotics), like distributed sensing tasks in micromachinery or the human body. One of the most promising uses of swarm robotics is in disaster rescue missions. Swarms of robots of different sizes could be sent to places rescue workers cannot reach safely, to detect the presence of life via infrared sensors. On the other hand, swarm robotics can be suited to tasks that demand cheap designs, for instance mining or agricultural foraging tasks. Also some artists use swarm robotic techniques to realize new forms of interactive art.

-Maniteja(B15CS063)

Flutter

The first version of Flutter was known as "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second.

The major components of Flutter include:

Flutter engine
Foundation library
Design-specific widgets

Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS.

Foundation library

The Foundation library, written in Dart, provides basic classes and functions which are used to construct applications using Flutter, such as APIs to communicate with the engine.

Widgets

UI design in Flutter involves assembling and/or creating various widgets. A widget in Flutter represents an immutable description of part of the user interface; all graphics, including text, shapes, and animations are created using widgets. More complex widgets can be created by combining many simpler ones.

Design-specific widgets

The Flutter framework contains two sets of widgets which conform to specific design languages. Material Design widgets implement Google's design language of the same name, and Cupertino widgets imitate Apple's iOS design.

-M. Sai Karthik (B15CS065)

Xamarin

Xamarin is a Microsoft-owned San Francisco, California-based software company founded in May 2011 by the engineers that created Mono, Mono for Android and MonoTouch, which are cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications (often called Microsoft .NET).

With a C# -shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows and macOS. According to Xamarin, over 1.4 million developers were using Xamarin's products in 120 countries around the world as of April 2017.

On February 24, 2016, Microsoft announced it had signed a definitive agreement to acquire Xamarin.

Origins in Ximian and Mono

In 1999 Miguel de Lacza and Nat Friedman launched what would eventually be known as Ximian to support and develop software for de Icaza's nascent GONME project. After Microsoft first announced their .NET Framework in June 2000,de Icaza began investigating whether a Linux version was feasible. The Mono open source project was launched on July 19, 2001. Ximian was bought by Novell on August 4, 2003, which was then acquired by Attachmate in April 2011.

After the acquisition, Attachmate announced hundreds of layoffs for the Novell workforce, including Mono developers, putting the future of Mono in question.

Founding Xamarin

On May 16, 2011, Miguel de Icaza announced on his blog that Mono would be developed and supported by Xamarin, a newly formed company that planned to release a new suite of mobile products. According to de Icaza, at least part of the original Mono team had moved to the new company.

The name Xamarin comes from the name of the Tamarin monkey, replacing the leading T with an X. This is in line with the naming theme used ever since Ximian was started.

After Xamarin was announced, the future of the project was questioned, since MonoTouch and Mono for Android would now be in direct competition with the existing commercial offerings owned by Attachmate. It was not known at that time how Xamarin would prove they had not illegally used technologies previously developed when they were employed by Novell for the same work.

In July 2011, however, Novell - now a subsidiary of Attachmate - and Xamarin announced that Novell had granted a perpetual license for Mono, MonoTouch and Mono for Android to Xamarin, which formally and legally took official stewardship of the project.

Product development

In December 2012, Xamarin released Xamarin.Mac, a plugin for the existing MonoDevelop Integrated development environment (IDE), which allows developers to build C#-based applications for the Apple OS X operating system and package them for publishing via the Apple App Store.

In February 2013, Xamarin announced the release of Xamarin 2.0.The release included two main components: Xamarin Studio, a re-branding of its open-source IDE Monodevelop;and integration with Visual Studio, Microsoft's IDE for the .NET Framework, allowing Visual Studio to be used for creating applications for Android and iOS, as well as for Windows.

Funding

On July 17, 2013 Xamarin announced that they had closed $16 million in Series B funding led by Lead Edge Capital.[22] Several investors from their Series A funding also participated, including Charles River Ventures, Floodgate, and Ignition Partners. On August 21, 2014 Xamarin successfully closed an additional $54 million in Series C funding, which is one of the largest rounds of funding ever raised by a mobile app development platform.Total funding for the company to date is $82 million.

Acquisition

On February 24, 2016 Xamarin and Microsoft announced that Microsoft signed a definitive agreement to acquire Xamarin.Terms of the deal weren't disclosed, though the Wall Street Journal reported the price at between $400 million and $500 million.

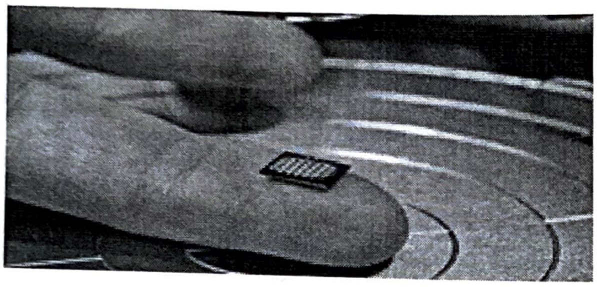
Microsoft subsidiary (2016-present)

At Microsoft Build 2016 Microsoft announced that they will open-source the Xamarin SDK and that they will bundle it as a free tool within Microsoft Visual Studio's integrated development environment, and Visual Studio Enterprise users would also get Xamarin's enterprise features free of charge. As a part of the acquisition they would also relicense Mono completely under the MIT License and would release all other Xamarin SDK software through the .NET Foundation also under the MIT License.

-M. Sai Karthik (B15CS065)

WORLD'S SMALLEST COMPUTER

IBM is kicking off its IBM Think 2018 conference this week with "5 in 5," a collection of IBM Research inventions and technologies "that could change our lives in the next five years." If you want to hear a large corporation tell you about AI, block chain, and quantum computing all in the same breath, IBM Think sounds like the place to be.



It's a little hard to tease out the technology from the buzzwords, but, happily, Mashable spotted this gem: IBM is building the world's smallest computer. Details are still thin - perhaps we'll learn more this week - but there's enough info to get excited about.

The computer is 1mm x 1mm, smaller than a grain of fancy salt, and apparently costs less than ten cents to manufacture. To be clear, the picture above is a set of 64 motherboards, each of which hold two of this tiny computer. Here's an actual photo of a solo computer on a pile of salt for scale:

In comparison, the last "world's smallest computer" to make a big splash was the Michigan Micro Mote in 2015, which measured a whopping 2mm across.

Feature-wise, the computer has a processor with "several hundred thousand" transistors, SRAM memory, a photo-voltaic cell for power, and a communications unit that uses an LED and a photo-detector to talk with the outside world.

IBM claims the computer has the power of an x86 chip from 1990. That puts it exactly on the edge of enough power to run the original Doom (the original README.TXT for Doom says a 386 processor and 4MB of RAM is the minimum). Hopefully IBM will be more forthcoming with benchmarks in the next five years, and I'm looking forward to repurposing this chip's LED as a one pixel display.

IBM's actual application for this chip seems mostly centered on supply chain management and conterfiet protection - enter the "blockchain" buzz. The chip is just one of many "crypto-anchors" IBM is developing for this purpose.

-Ranjith Katta(B15CS042)

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING